Semantic Gaussian Mixture Variational Autoencoder for Sequential Recommendation*

Beibei Li[1], Tao Xiang ✉[1], Beihong Jin ✉[2,3], Yiyuan Zheng[2,3], and Rui Zhao[2,3]

[1] College of Computer Science, Chongqing University, Chongqing, China
{libeibeics,txiang}@cqu.edu.cn

[2] Key Laboratory of System Software (Chinese Academy of Sciences) and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Science, Beijing, China Beihong@iscas.ac.cn

[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Variational AutoEncoder (VAE) for Sequential Recommendation (SR), which learns a continuous distribution for each user-item interaction sequence rather than a determinate embedding, is robust against data deficiency and achieves significant performance. However, existing VAE-based SR models assume a unimodal Gaussian distribution as the prior distribution of sequence representations, leading to restricted capability to capture complex user interests and limiting recommendation performance when users have more than one interest. Due to that it is common for users to have multiple disparate interests, we argue that it is more reasonable to establish a multimodal prior distribution in SR scenarios instead of a unimodal one. Therefore, in this paper, we propose a novel VAE-based SR model named SIGMA. SIGMA assumes that the prior of sequence representation conforms to a Gaussian mixture distribution, where each component of the distribution semantically corresponds to one of a user's multiple interests. For multi-interest elicitation, SIGMA includes a probabilistic multi-interest extraction module that learns a unimodal Gaussian distribution for each interest according to implicit item hyper-categories. Additionally, to incorporate the multimodal interests into sequence representation learning, SIGMA constructs a multi-interest-aware ELBO, which is compatible with the Gaussian mixture prior. Extensive experiments on public datasets demonstrate the effectiveness of SIGMA. The code is available at https://github.com/libeibei95/SIGMA.

1 Introduction

Sequential Recommendation (SR) [15, 13] aims to predict the next item to be interacted with according to a user's history interaction sequence. By learning the evolving trends of user interests over time and the transition between

* This work was supported by Chongqing Science and Technology Bureau (CSTB2022TIAD-KPX0180) and the National Natural Science Foundation of China under Grant No. 62072450.

items, SR models can capture user interests and predict target items accurately. However, most of existing SR models learn determinate representations for sequences, which lack smoothness, i.e., small perturbations of embeddings can lead to totally different recommendations. Therefore, they are easy to be deteriorated by the common data sparsity and data noise problems, which disturb sequence embedding learning and have not been well-solved. To address this issue, Variantial AutoEncoder (VAE) [5, 8], a representative probabilistic latent variable model, is introduced into SR [9, 19] and represents sequences with distributions rather than embeddings, so that the representation continuity is guaranteed. VAE-based SR models combine the strong representation power of latent spaces provided by VAE, with the sequence modeling capabilities of deep learning, achieving significant performance improvement.

Unfortunately, conventional VAE suffers from posterior collapse caused by over-regularization [16], i.e., the estimates posterior distributions of different input data are undistinguishable in the latent space due to that they are all optimized towards the standard Gaussian distribution. VAE-based SR models [9] inherit this problem. To resolve the problem, adversarial training and contrastive learning[17, 14] are introduced into VAE-based SR, enhancing the quality of latent variables and achieving substantial performance improvement. Nevertheless, these models assume that sequence representations conform to the unimodal prior distributions, which exhibit limitations in capturing complex and diverse user interests. In real-world SR scenarios, it is common for a user's interaction sequence to involve multiple interests. Therefore, it is more reasonable to assume that the sequence representations are generated from a multimodal latent space rather than a unimodal one.

To enhance the representation capability of the latent space in VAE-based SR, we propose a novel VAE model named **SIGMA** (**S**emant**I**c **G**aussian **M**ixture variational **A**utoencoder) in this paper. Unlike existing models, SIGMA assumes the prior distribution of sequence representation is multimodal and follows a semantic Gaussian mixture distribution, where each component corresponds to one of a user's multiple user interests. Specifically, SIGMA establishes an Evidence Lower BOund (ELBO) that is compatible with the mixture Gaussian prior for SR. Then, to estimate the distribution of each user interest, SIGMA incorporates a VAE-based multi-interest extraction module, which disentangles the user's multiple interests from the interaction sequence and learns a unimodal Gaussian distribution for each individual interest. Finally, extensive experiments on three public datasets are conducted and the results demonstrate that the SIGMA outperforms both representative SR models and multi-interest recommendation models.

2 Preliminaries

2.1 Problem Definition

We denote the user set as $\mathcal{U}$, the item set as $\mathcal{V}$, and the embedding of an item $v \in \mathcal{V}$ as $\boldsymbol{v} \in \mathbb{R}^d$, where d is the dimension of embeddings. For a user $u \in \mathcal{U}$, we sort his/her interactions in ascending order by timestamp, obtaining an interaction sequence $\boldsymbol{s} = [v_1, v_2, \ldots, v_T]$. Our goal is to build a model that predicts the next item that the user u_i is most likely to interact with at the $T+1$ step among the item set $\mathcal{V}$ given the sequence $\boldsymbol{s}$ as input $\operatorname{argmax}_{v_j \in \mathcal{V}} P\left(v_{T+1} = v \mid \boldsymbol{s}\right)$.

2.2 ELBO for SR

Given a user interaction sequence $\boldsymbol{s} = [v_1, v_2, \ldots, v_T]$, where T is the length of the interaction sequence, we assume its generative process is as follows $\prod_{t=0}^{T-1} p(v_{t+1} \mid \boldsymbol{z}_t) = \prod_{t=0}^{T-1} p_{\theta'}(v_{t+1} \mid \boldsymbol{z}_t) p(\boldsymbol{z}_t)$, where $\boldsymbol{z}_t \in \mathbb{R}^d$ is the latent variable that generates the next item v_{t+1}, $p(\boldsymbol{z}_t)$ represents the prior probability of $\boldsymbol{z}_t$, and $p_{\theta'}(v_{t+1} \mid \boldsymbol{z}_t)$ represents the likelihood parameterized by θ'. The log-likelihood of the user sequence is expressed as follows:

$$\log \prod_{t=1}^{T} p(v_t) = \sum_{t=1}^{T} \log p(v_t) = \sum_{t=0}^{T-1} \log \int_{\boldsymbol{z}_t} p(\boldsymbol{z}_t) p_{\theta'}(v_{t+1} | \boldsymbol{z}_t) d\boldsymbol{z}_t.$$

Maximizing the above logarithmic likelihood is intractable since it is necessary to calculate all possible latent variables $\boldsymbol{z}$. To address this issue, VAE-based SR models follow the standard VAE to leverage variational inference and optimize the model by maximizing the evidence lower bound (ELBO), which consists of a reconstruction term and a KL-divergence term. Therefore, the ELBO for VAE-based SR can be written in the following form:

$$ELBO := \sum_{t=0}^{T-1} (\underbrace{\mathbb{E}_{q_{\phi'}(\boldsymbol{z}_t | \boldsymbol{s}_{1:t})} \left[\log p_{\theta'}(v_{t+1} \mid \boldsymbol{z}_t)\right]}_{\text{Reconstruction term } \mathcal{T}_{recon}} - \underbrace{D_{KL}\left[q_{\phi'}(\boldsymbol{z}_t \mid \boldsymbol{s}_{1:t}) \| p(\boldsymbol{z}_t)\right]}_{\text{KL-divergence term } \mathcal{T}_{KL}}),$$

where $\mathcal{T}_{recon}$ and $\mathcal{T}_{KL}$ denote the reconstruction term and KL-divergence term, respectively. $\boldsymbol{s}_{1:t}$ represents the prefix subsequence consisting of the first t items of sequence $\boldsymbol{s}$, $q_{\phi'}(\boldsymbol{z}_t \mid \boldsymbol{s}_{1:t})$ is the estimated posterior probability, and $q_{\phi'}(\cdot)$ and $p_{\theta'}(\cdot)$ are the encoder and decoder built on neural networks, respectively. In SR scenarios, the encoder and decoder are usually instantiated with sequence modeling methods such as RNN and Transformer.

3 Methodology

3.1 Overview

SIGMA consists of a Multi-Interest Extraction VAE (MIE-VAE) and a Semantic Gaussian Mixture VAE (SGM-VAE). The former focuses on interest representation learning, while the latter is designed for sequence representation learning under the assumption of multi-modal prior distribution. Specifically, MIE-VAE aims to disentangle multiple interests from each sequence, represent each interest with a unimodal Gaussian distribution and quantify the intensity of each interest. The weighted mixture of these interest distributions forms a semantic Gaussian mixture distribution and serves as the prior distribution for sequence representation. SGM-VAE learns the posterior distribution of the sequence representation and aligns it with the Gaussian mixture distribution provided by MIE-VAE using KL divergence.

3.2 Multi-Interest Extraction VAE (MIE-VAE)

MIE-VAE aims to extract multiple interest representations $\mathcal{X}_t$ from the user sequence $\boldsymbol{s}_{1:t}$, where $\mathcal{X}_t = [\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \cdots, \boldsymbol{x}_t^k]$ denotes the k user interests extracted from the prefix subsequence $\boldsymbol{s}_{1:t}$, $\boldsymbol{x}_t^i$ follows a unimodal Gaussian distribution, that is, $\boldsymbol{x}_t^i \sim \mathcal{N}(\boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I})$, $\boldsymbol{m}_t^i$ and $\boldsymbol{\omega}_t^i$ respectively denote the mean and standard variance of the i-th interest, and $\boldsymbol{I} \in \mathbb{R}^{k \times k}$ is an identity matrix. The representation of the i-th interest, i.e. $\boldsymbol{x}_t^i$, is sampled from the distribution $\mathcal{N}\left(\boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I}\right)$, that is, $\boldsymbol{x}_t^i = \boldsymbol{m}_t^i + \boldsymbol{\omega}_t^i \odot \boldsymbol{\epsilon}$.

We construct an encoder to estimate the posterior probability of interest representations given the sequence $\boldsymbol{s}_{1:t}$, as follows:

$$\begin{aligned} q_\phi(\mathcal{X}_t \mid \boldsymbol{s}_{1:t}) &= q_\phi\left([\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, \cdots, \boldsymbol{x}_t^k] \mid \boldsymbol{s}_{1:t}\right) \\ &= \prod\nolimits_{i=1}^k q_\phi\left(\boldsymbol{x}_t^i \mid \boldsymbol{s}_{1:t}\right) \sim \prod\nolimits_{i=1}^k \mathcal{N}\left(\boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I}\right), \qquad (1) \\ [\boldsymbol{m}_t^1, \boldsymbol{m}_t^2, \ldots, \boldsymbol{m}_t^k] &= \mathrm{MultiEncoder}_m(\boldsymbol{s}_{1:t}), \qquad (2) \\ [\boldsymbol{\omega}_t^1, \boldsymbol{\omega}_t^2, \ldots, \boldsymbol{\omega}_t^k] &= \mathrm{MultiEncoder}_\omega(\boldsymbol{s}_{1:t}), \qquad (3) \end{aligned}$$

where ϕ denotes the parameters in the constructed encoder. In addition, $\mathrm{MultiEncoder}_m(\cdot)$ and $\mathrm{MultiEncoder}_\omega(\cdot)$ are encoders constructed to compute the mean and standard deviation to represent interests, which are introduced as follows.

Multi-Interest Encoder Items can be divided into several categories, such as health, education, etc. Each category corresponds to one user interest. Therefore, we rely on implicit item categories to learn user interests.

To mine the item categories, we set up k global category embeddings $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_k]$, where $\boldsymbol{g}_* \in \mathbb{R}^d$ and 2-norm normalized. To reduce redundancy among different categories, we ensure that the global category embeddings are pairwise orthogonal. Therefore, we construct the following orthogonal constraint loss: $\mathcal{L}_{orth} = \sum_{i=1}^k \sum_{j=1, j \neq i}^k \boldsymbol{g}_i^T \boldsymbol{g}_j$.

We calculate the correlation score between v_i and the k-th category using $\boldsymbol{g}_k^T \boldsymbol{v}_i$, where $\boldsymbol{v}_i$ is normalized to the 2-norm. Further, we utilize correlation scores to compute the classification probability of items into each category via softmax function. For example, the probability of v_i belonging to the j-th category can be calculated as $a_i^j = \frac{\exp\left(\boldsymbol{g}_j^T \boldsymbol{v}_i / \tau\right)}{\sum_{l=1}^k \exp\left(\boldsymbol{g}_l^T \boldsymbol{v}_i / \tau\right)}$, where τ is a temperature coefficient. The larger τ, the smoother the probability values.

With the classification probabilities, we learn both **soft interests** and **hard interests** via two different strategies, where soft interest allows each item to be related to multiple user interests, while hard interests restrict each item to be exclusively related to only one user interest.

For interaction sequence $\boldsymbol{s}_{1:t}$, the j-th soft interests $\boldsymbol{h}_t^j$ is calculated with the weighted summation of embeddings of his/her interacted items classified into the category i, as $\boldsymbol{h}_t^j = \sum_{i=0}^t a_i^j \boldsymbol{v}_i$. The intensity of the j-th interest implied in sequence $\boldsymbol{s}_{1:t}$ is calculated as $\alpha_t^j = \frac{\sum_{i=1}^t a_i^j}{\sum_{i=1}^t \sum_{j=1}^k a_i^j}$. Interest intensity is employed as the weight of the corresponding interest distribution in the semantic Gaussian mixture prior, which will be illustrated in Section 3.3.

To complement that soft interests cannot capture the evolution of user interests over time, we also learn hard interests for users. We initially assign each interacted item to a category exclusively and then model the evolution of each user interest over time using sequence models. Naturally, we are supposed to assign each interacted item v_i to the category l with the highest classification probability, i.e., $l = \mathrm{argmax}_{1\leq j \leq k}\left(\{a_{ij}\}\right)$. According to the hard allocation strategy, the interaction sequence $\boldsymbol{s}$ is split into k disjoint subsequences s_j, where $1 \leq j \leq k$, $\sum_{j=1}^{k} |\boldsymbol{s}_j| = |\boldsymbol{s}|$. The number of subsequences with non-zero length is the number of hard interests, which is dynamic and personalized. Furthermore, to model the temporal evolution of each user's interest, for each non-empty subsequence, we utilize the sequential model to capture the changes in user interest and generate the corresponding hard interest vector. Here, we leverage GRU [15]. The i-th hard interest associated with sequence $\boldsymbol{s}_{1:t}$ can be denoted as $\boldsymbol{r}_t^i$.

By combining the soft interests and hard interests mentioned above, we compute the mean vector of the i-th interest for user u, represented as $\boldsymbol{m}_t^i = (\boldsymbol{h}_t^i + \boldsymbol{r}_t^i)/2$. As for the standard deviation $\boldsymbol{\omega}_i$ of the i-th interest, we employ multi-layer perceptrons with the concatenation of the hyper-parameter embedding $\boldsymbol{g}_i$ and the mean vector of user interests $\boldsymbol{m}_t^i$ as input, and calculate it as $\boldsymbol{\omega}_t^i = \mathrm{MLP}([\boldsymbol{g}_i \| \boldsymbol{m}_t^i])$.

ELBO for Multi-interest Recommendation For the KL-divergence term, we assume the prior $p(\mathcal{X}_t) = \prod_{i=1}^{k} p(\boldsymbol{x}_t^i)$ and $p(\boldsymbol{x}_t^i) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for all interests. Combining with the estimated posterior for i-th interest of user u is another Gaussian distribution $\mathcal{N}(\boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I})$, the closed-form solution of the KL-divergence could be easily computed through:

$$\begin{aligned}\mathcal{T}_{KL}^{MIE} &= \sum\nolimits_{t=1}^{T} D_{KL}[q_\phi(\mathcal{X}_t \mid \boldsymbol{s}_{1:t}) \| p_\theta(\mathcal{X}_t)] \\ &= \sum\nolimits_{t=1}^{T} \sum\nolimits_{i=1}^{k} \sum\nolimits_{j=1}^{d} \left(\omega_{t,j}^{i2} + m_{t,j}^{i2} - 1 - \log \omega_{t,j}^{i2}\right). \end{aligned} \tag{4}$$

For the reconstruction loss, similar to the previous section, we construct it according to the next-item prediction task, as follows:

$$\mathcal{T}_{recon}^{MIE} = \sum\nolimits_{t=1}^{T} \mathbb{E}_{q_\phi(\mathcal{X}_t | \boldsymbol{s}_{1:t})} \log[p_\theta(v_{(t+1)} \mid \mathcal{X}_t)]. \tag{5}$$

We map the j-th interest embedding extracted from $\boldsymbol{s}_{1:t}$, i.e., $\boldsymbol{x}_t^j$ sampled from $\mathcal{N}\left(\boldsymbol{m}_t^j, {\boldsymbol{\omega}_t^j}^2 \boldsymbol{I}\right)$ to the same space as the item embeddings using two-layer perceptrons, i.e., $\boldsymbol{u}_t^j = \mathrm{MLP}(\boldsymbol{x}_t^j)$. For a candidate item $v \in \mathcal{V}$, we calculate its relevance scores with each user's interest. For example, the relevance score with the i-th interest is ${\boldsymbol{u}_t^j}^T \boldsymbol{v}$. Following previous works on multi-interest recommendation [6], we consider the maximum relevance score between v and the reconstructed interest representation as the final prediction score. We calculate the interaction probability as $p(v_{T+1} = v) = \frac{e^{\max\left(\left\{<\boldsymbol{u}_T^j, \boldsymbol{v}>/\epsilon | 1 \leq j \leq k\right\}\right)}}{\sum_{v' \in \mathcal{V}} e^{\max\left(\left\{<\boldsymbol{u}_T^j, \boldsymbol{v'}>/\epsilon | 1 \leq j \leq k\right\}\right)}}$.

Note that we learn user interest distribution on each t, which is equivalent to augmenting training data by truncating prefix sequences. While benefiting representation learning, these interests can be calculated in parallel via optimized matrix calculations, resulting in no additional time consumption.

3.3 Semantic Gaussian Mixture VAE (SGM-VAE)

Ideal representations of interaction sequences are supposed to reflect the multiple interests of users. In SGM-VAE, we aim to capture complex and diverse user interests by establishing a multimodal prior distribution for sequence representation. To achieve this, we establish an ELBO compatible with semantic Gaussian mixture distribution, which consists of two components: a KL-divergence term and a reconstruction loss.

Semantic Gaussian Mixture KL-divergence Considering that users usually have multiple interests and each interest can be represented by a unimodal Gaussian distribution, naturally, the prior distribution of sequence representation can be assumed to follow Gaussian mixture distributions composed of multiple interests. Given the distribution of interests learned via MIE-VAE from an interaction sequence $\boldsymbol{s}$, the prior distribution of the sequence representation can be represented as the following Gaussian mixture distribution, $p(\boldsymbol{z}_t) = \sum_{i=1}^{k} \alpha_t^i \mathcal{N}(\boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I})$, where α_t^i represents the intensity of the i-th interest implied by the sequence $\boldsymbol{s}_{1:t}$. Since the representations of different interests are personalized and convey different semantics, we can depend on these interests to obtain a distinct semantic prior for each interaction sequence so as to avoid posterior collapse.

The posterior probability distribution of the latent variable $\boldsymbol{z}_t$ for sequence $\boldsymbol{s}_{1:t}$ can be represented as $q_{\phi'}(\boldsymbol{z}_t \mid \boldsymbol{s}_{1:t}) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I})$, where $\boldsymbol{\mu}_t = Enc_\mu(\boldsymbol{s}_{1:t})$ and $\boldsymbol{\sigma}_t = Enc_\sigma(\boldsymbol{s}_{1:t})$. $Enc_\mu(\cdot)$ and $Enc_\sigma(\cdot)$ are Transformer encoders used to compute the mean and standard deviation of the Gaussian distribution for sequence representation, respectively.

The KL divergence between the estimated posterior and the Gaussian mixture prior cannot be directly computed analytically like the standard Gaussian prior. Therefore, similar to [11], we approximate the KL divergence using the following equation:

$$\begin{aligned} \mathcal{T}_{KL}^{SGM} &\approx \sum_{t=1}^{T} (\log q_{\phi'}(\boldsymbol{z}_t \mid \boldsymbol{s}_{1:t}) - \log p(\boldsymbol{z}_t)) \\ &= \sum_{t=1}^{T} (\log \mathcal{N}(\boldsymbol{z}_t \mid \boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I}) - \log \sum_{i=1}^{k} \alpha_t^i \mathcal{N}(\boldsymbol{z}_t \mid \boldsymbol{m}_t^i, {\boldsymbol{\omega}_t^i}^2 \boldsymbol{I})), \end{aligned} \tag{6}$$

where α_t^i is the interest intensity learned from $\boldsymbol{s}_{1:t}$ and introduced in Section 3.2.

Reconstruction Term Since the task of SR is to predict the next item of interest, the generative process is as follows. We first sample an embedding from the estimated posterior probability $q_{\phi'}(\boldsymbol{z}_t \mid \boldsymbol{s}_{1:t}) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I})$ using the reparameterization trick, where $\boldsymbol{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is sampled from a standard Gaussian distribution. Next, we construct a decoder based on the Transformer that is almost identical to the encoder structure to predict $p(v_{t+1} \mid \boldsymbol{z}_t)$. Let $\hat{\boldsymbol{u}}_t$ denote the output of the decoder at position t. Regarding the next-item prediction problem as a multi-classification problem, we calculate the interaction score between $\hat{\boldsymbol{u}}_t$ and each candidate item $v \in \mathcal{V}$ sequentially and convert it into an interaction probability using the softmax function. The $p(v_{t+1} \mid \boldsymbol{z}_t)$ is calculated as follows, $p(v_{t+1} = v \mid \boldsymbol{z}_t) = \frac{\exp(\hat{\boldsymbol{u}}_t^T \boldsymbol{v}/\tau)}{\sum_{v' \in \mathcal{V}} \exp(\hat{\boldsymbol{u}}_t^T \boldsymbol{v}'/\tau)}$.

Then, we optimize the next-item prediction task by a reconstruction term as follows:

$$\mathcal{T}^{SGM}_{recon} = \sum_{t=0}^{T-1} \mathbb{E}_{q_{\phi'}(\boldsymbol{z}_t|\boldsymbol{s}_{1:t})} \log[p(v_{t+1} \mid \boldsymbol{z}_t)]. \tag{7}$$

3.4 Training and Inference

During training, both the SGM-VAE and MIE-VAE are simultaneously fed with interaction sequences. We jointly the two VAEs by constructing the following loss function:

$$\mathcal{L} = -\underbrace{(\mathcal{T}^{SGM}_{recon} - \lambda \mathcal{T}^{SGM}_{KL})}_{\text{ELBO of SGM-VAE}} - \beta_1 \underbrace{(\mathcal{T}^{MIE}_{recon} - \mathcal{T}^{MIE}_{KL})}_{\text{ELBO of MIE-VAE}} + \underbrace{\beta_2 \mathcal{L}_{orth}}_{\text{Regularization}} ,$$

where $\lambda, \beta_1, \beta_2$ are hyperparameters. λ is introduced as weight for the KL term to avoid posterior collapse and over-regularization [10].

As for inference, we learn the posterior distribution $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\sigma}_t^2 \boldsymbol{I})$ of input sequences with SGM-VAE first. Then, we use $\boldsymbol{\mu}_t$ to calculate the user representation $\hat{\boldsymbol{u}}_t$ and generate the final recommendation.

4 Experiments

4.1 Experimental Setting

Datasets We adopt four publicly available datasets, including three Amazon datasets, namely *Beauty*, *Toys and Games* (*Toys* for short), and *Office Product* (*Office* for short), and *MovieLens*. We follow the common settings in previous settings [18].

Metrics Recall@K and NDCG@K are used as metrics to evaluate the performance of the recommendation results. We report the results where $K = 20$ and $K = 40$.

Compared Models We compare SIGMA to several representative models, which can be separated into four classes. 1) Collaborative filtering: *BPRMF* [7] and *LightGCN* [3]. 2) Attention-based SR models: *SASRec* [4] and *Bert4Rec* [12]. 3) VAE-based SR models: *SVAE* [9], *ACVAE* [17] and *DT4SR* [2]. 4) Multi-interest recommendation models: *MIND* [6], *ComiDR* and *ComiSA*[1].

Table 1. Recommendation performance on three datasets, where R denotes Recall and N denotes NDCG. We bold the best results and underline the second best results of each metric. The last column is the relative improvements compared with the best baseline results.

	Beauty				**Office**				**Toys**			
Models	R@20	R@40	N@20	N@40	R@20	R@40	N@20	N@40	R@20	R@40	N@20	N@40
BPRMF	0.0739	0.1089	0.0311	0.0383	0.0483	0.0718	0.0218	0.0266	0.0692	0.1007	0.0304	0.0369
LightGCN	0.0759	0.1112	0.0306	0.0378	0.0532	0.0797	0.0243	0.0297	0.0671	0.0977	0.0287	0.0349
Bert4Rec	0.0890	0.1285	0.0395	0.0476	0.1350	0.2230	0.0551	0.0729	0.0699	0.0982	0.0318	0.0376
SASRec	0.0952	0.1389	0.0420	0.0509	0.1478	0.2251	0.0657	0.0815	0.1112	0.1479	0.0539	0.0614
SVAE	0.0268	0.0417	0.0102	0.0132	0.0988	0.1647	0.0389	0.0523	0.0178	0.0260	0.0069	0.0086
ACVAE	0.0951	0.1294	0.0467	0.0537	0.1327	0.2075	0.0560	0.0713	0.0722	0.1030	0.0359	0.0421
DT4SR	0.0982	0.1404	0.0446	0.0533	0.1429	0.2186	0.0643	0.0797	0.1130	0.1478	0.0515	0.0560
MIND	0.0588	0.0859	0.0247	0.0297	0.0614	0.1066	0.0240	0.0354	0.0555	0.0801	0.0243	0.0289
ComiRec-DR	0.0373	0.0583	0.0148	0.0194	0.0263	0.0601	0.0106	0.0178	0.0180	0.0266	0.0080	0.0097
ComiRec-SA	0.0574	0.0906	0.0200	0.0271	0.0685	0.1193	0.0233	0.0335	0.0474	0.0722	0.0179	0.0228
SIGMA	**0.1048**	**0.1494**	**0.0477**	**0.0568**	**0.1615**	**0.2334**	**0.0722**	**0.0869**	**0.1151**	**0.1510**	**0.0549**	**0.0629**
Improv.	6.69%	6.41%	2.13%	5.77%	9.25%	3.70%	9.95%	6.61%	1.86%	2.09%	1.94%	2.38%

4.2 Performance Comparison

The experimental results are shown in Table 1, from which we have the following observations.

Firstly, the quality of posterior probability estimation plays a crucial role in determining the recommendation performance of VAE-based SR models. When compared to other competitors, such as ACVAE and DT4SR, SVAE exhibits extremely poor performance. This can be attributed to the fact that SVAE adheres to the conventional VAE approach, which assumes that the embeddings of each user are sampled from a standard Gaussian distribution. The similarity between the assumed prior distribution and the estimated posterior distribution results in the collapse of user representations, leading to the loss of personalized information. This phenomenon is particularly pronounced when the number of items is large, as observed in datasets such as Beauty and Toy. ACVAE and DT4SR, which enhance the inference of latent variables by employing the adversarial variational Bayes technique and utilizing Elliptical Gaussian distributions, respectively, achieve significant performance improvement compared to SVAE.

Moreover, existing multi-interest recommendation models that acquire multiple interest representations for users have not exhibited superior performance. This discrepancy in the expectation arises from two main factors. Firstly, unlike other models that solely prioritize recommendation accuracy, multi-interest recommendation models also need to take into account the diversity of recommendations. Consequently, these two objectives may lead to an optimization dilemma. Secondly, multi-interest models constructed with dynamic routing, such as MIND and ComiRec-DR, do not consider the sequential nature of a user's history interactions. Instead, they treat these interactions as an unordered item set.

Finally, the proposed model SIGMA outperforms all competitors, highlighting its effectiveness. This superiority can be attributed to the semantic and expressive sequence representation learned by SIGMA. Notably, SIGMA demon-

Table 2. Ablation Study

Datasets	**w/o MIE-VAE**		**w/o SGM-VAE**		**w/o Orth.**		**SIGMA**	
	R@40	N@40	R@40	N@40	R@40	N@40	R@40	N@40
Beauty	0.1333	0.0523	0.1315	0.0469	0.1415	0.0517	**0.1494**	**0.0568**
Office	0.2243	0.0828	0.1806	0.0651	0.1827	0.0640	**0.2334**	**0.0869**
Toy	0.1439	0.0601	0.1319	0.0552	0.1455	0.0579	**0.1510**	**0.0629**

strates the greatest improvement on the Office dataset that comprises longer average interaction sequences than those of the other two datasets, which means that SIGMA is more adept at learning high-quality interest representations from longer sequences

4.3 Ablation Study

In this section, we conduct ablation studies to investigate the impact of different components in our SIGMA model, including the MIE-VAE, SGM-VAE, and the orthogonal constraint $\mathcal{L}_{orth}$.

For the MIE-VAE, we compared SIGMA with its variant SIGMA$_{uni}$. As shown in Table 2, the performance of SIGMA$_{uni}$ falls far behind SIGMA. For example, on Beauty, Recall@40 and NDCG@40 dropped by 10.75% and 7.99% respectively, highlighting the value of the Gaussian Mixture prior based on multiple interests. As for the SGM-VAE, we train MIE-VAE separately and used it for recommendation. there was a sharp decline in performance as shown in Table 2. It indicates that SGM-VAE plays a crucial role in integrating users' diverse interests and considering their varying intensities. Regarding the orthogonal constraint, we remove $\mathcal{L}_{orth}$ and find a significant decrease in the accuracy of the recommendation, especially on the Office dataset. This confirms the necessity of the orthogonal constrain for learning semantic category representations and enhancing recommendation quality. Overall, it can be seen that no matter which part is removed, the performance will decline, which shows the important roles of different components.

5 Conclusion

In this paper, we propose adopting a Gaussian mixture prior distribution to describe user interests implied in each interaction sequence and build a novel variational autoencoder for SR, named SIGMA. Specifically, we construct a multi-interest extraction VAE that learns a unimodal Gaussian distribution for each interest and mixes them based on the interest intensity. By incorporating a KL divergence between the estimated posterior distribution of sequence representation and the Gaussian mixture distribution composed of multiple interests, SIGMA enables the sequence representation to capture the complex and diverse preferences of users. Experimental results show that SIGMA surpasses existing SR methods, demonstrating the contribution of the semantic Gaussian mixture prior to performance improvement.

References

1. Cen, Y., Zhang, J., Zou, X., Zhou, C., Yang, H., Tang, J.: Controllable multi-interest framework for recommendation. In: SIGKDD. p. 2942–2951 (2020)
2. Fan, Z., Liu, Z., Wang, S., Zheng, L., Yu, P.S.: Modeling sequences as distributions with uncertainty for sequential recommendation. In: CIKM. p. 3019–3023 (2021)
3. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: SIGIR. pp. 639–648 (2020)
4. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: ICDM. pp. 197–206 (2018)
5. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
6. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L.: Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In: CIKM. pp. 2615–2623 (2019)
7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: UAI. p. 452–461 (2009)
8. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: ICML. p. II–1278–II–1286 (2014)
9. Sachdeva, N., Manco, G., Ritacco, E., Pudi, V.: Sequential variational autoencoders for collaborative filtering. In: WSDM. p. 600–608 (2019)
10. Shenbin, I., Alekseev, A., Tutubalina, E., Malykh, V., Nikolenko, S.I.: Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In: WSDM. p. 528–536. ACM
11. Shu, R.: Gaussian mixture vae: Lessons in variational inference, generative models, and deep nets (2016)
12. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: CIKM. p. 1441–1450 (2019)
13. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. pp. 17–22 (2016)
14. Wang, Y., Zhang, H., Liu, Z., Yang, L., Yu, P.S.: Contrastvae: Contrastive variational autoencoder for sequential recommendation. In: CIKM. p. 2056–2066 (2022)
15. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: WSDM. pp. 495–503 (2017)
16. Wu, M., Goodman, N.: Multimodal generative models for scalable weakly-supervised learning. NeurIPS **31** (2018)
17. Xie, Z., Liu, C., Zhang, Y., Lu, H., Wang, D., Ding, Y.: Adversarial and contrastive variational autoencoder for sequential recommendation. In: WWW. p. 449–459 (2021)
18. Zhang, S., Yang, L., Yao, D., Lu, Y., Feng, F., Zhao, Z., Chua, T.s., Wu, F.: Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation. In: WWW 2022. p. 2216–2226 (2022)
19. Zhao, J., Zhao, P., Zhao, L., Liu, Y., Sheng, V.S., Zhou, X.: Variational self-attention network for sequential recommendation. In: ICDE. pp. 1559–1570 (2021)