Stochastic-Expert Variational Autoencoder for Collaborative Filtering

Yoon-Sik Cho
Chung-Ang University
Seoul, Republic of Korea
yoonsik@cau.ac.kr

Min-hwan Oh
Seoul National University
Seoul, Republic of Korea
minoh@snu.ac.kr

ABSTRACT

Motivated by the recent successes of deep generative models used for collaborative filtering, we propose a novel framework of VAE for collaborative filtering using multiple experts and stochastic expert selection, which allows the model to learn a richer and more complex latent representation of user preferences. In our method, individual experts are sampled stochastically at each user-item interaction which can effectively utilize the variability among multiple experts. While we propose this framework in the context of collaborative filtering, the proposed *stochastic expert* technique can be used to enhance VAEs in general beyond the application of collaborative filtering. Hence, this novel technique can be of independent interest. We comprehensively evaluate our proposed method, *Stochastic-Expert Variational Autoencoder* (SE-VAE) on numerical experiments on the real-world benchmark datasets from MovieLens and Netflix and show that it consistently outperforms the existing state-of-the-art methods across all metrics. Our proposed stochastic expert framework is generic and adaptable to any VAE architecture. The experimental results show that the adaptations to various architectures provided performance gains over the existing methods.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; **Latent variable models**; **Neural networks**; • **Information systems** → **Recommender systems**.

KEYWORDS

Collaborative Filtering, Variational Autoencoder, Deep Generative Models, Neural Networks, Recommender Systems, Variational Inference

ACM Reference Format:
Yoon-Sik Cho and Min-hwan Oh. 2022. Stochastic-Expert Variational Autoencoder for Collaborative Filtering. In *WWW '22: Web Conference 2022, April 25—29, 2022, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485447.3512120

WWW '22, April 25—29, 2022, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512120

1 INTRODUCTION

Recommender systems are some of the most widely applied human-AI interactions today, one of the most active application domains where machine learning techniques are used. In essence, the recommender systems cater useful information and contents of potential interests to users without the users having to search for suitable contents. The (offline) recommendation problem[1] can be formalized as a matrix completion problem where user-item interactions (e.g., ratings, clicks, purchases) are recorded in a matrix but there are missing entries for which interactions have been observed. Hence, the objective of the matrix completion problem is to accurately predict missing entries of the matrix based on the observed values.

One of the key characteristics that an effective recommender systems should have is flexibility and expressiveness of the model to capture complex user preferences and interests. Collaborative filtering methods [19, 22] are some of the mostly widely used techniques. Due to the simplicity and tractability, latent variable models using matrix factorization [1, 3, 6, 16] have been a prevalent approach. Extending to more general function class, there has been an increasing body of literature proposing the adaptation of deep neural networks (DNN) to collaborative filtering in order to exploit their expressive power to account for non-linear and potentially more complex user-item preferences. Variational autoencoders (VAE) [10] have been proposed as a non-linear extension of classical latent variable models. The method proposed in [14] and its variants [9, 14, 20, 24, 25] using VAEs have been shown to significantly outperform the classical latent variable models based on matrix factorization. To our knowledge, VAE-based methods are currently the state-of-the-art in terms of the predictive performances in collaborative filtering.

However, the question of whether these methods are able to effectively learn richer and complex latent representation of user preferences still remains. The common issue that arises in VAEs for collaborative filtering is that the prior (and as a result also the posterior) distributions may be too simple and restrictive to learn potentially rich and complex latent representation of user preferences. There has been an effort to address this issue by allowing more flexible prior distributions. [9] proposed a VAE with a mixture distribution [23] (also knwon as *mixture-of-experts*) adapted to collaborative filtering. However, mixture procedure over multiple experts may not be able to fully exploit the variability across the multiple experts, hence not able to provide sufficient performance gains despite the potential expressive power given by the multiple experts (see more discussions in Section 2.4).

[1] Here we differentiate the *offline* recommendation problem from the *online* recommendation problem. The offline setting is where the historical batch data is available for learners, whereas in the online setting such data arrives sequentially.

To this end, we propose a novel VAE model for collaborative filtering, which we call the *Stochastic-Expert Variational Autoencoder* or SE-VAE for short. As the naming suggests, our model considers a set of multiple experts, where each expert is associated with its own deep latent Gaussian model. What differentiates our method from the existing methods using multiple experts is that, instead of utilizing the multiple experts as a mixture aggregation, we incorporate *stochasticity in expert selection*. That is, generating each user-item interaction sample, we randomly select an expert among the expert pool we maintain. This allows the model to incorporate diversity and flexibility in user's preferences, fully utilizing the variability created by multiple experts. We summarize the main contributions of our work as follows:

- We propose a novel framework of VAE for collaborative filtering using multiple experts and stochastic expert selection, which allows the model to learn richer and complex latent representation of user preferences.
- While we propose this framework in the context of collaborative filtering, the proposed *stochastic expert* technique can be used to enhance VAEs in general beyond the application of collaborative filtering. Hence, this novel technique can be of independent interest.
- We comprehensively evaluate our method SE-VAE on numerical experiments and show that it consistently outperforms the existing state-of-the-art methods across all metrics.
- Our proposed stochastic expert framework is generic and adaptable to any VAE architecture. The experimental results in Section 5 show that the adaptations to various architectures provided performance gains over the existing methods (see Table 2).

2 PRELIMINARIES

2.1 Problem Formulation

We are interested in the problem setting where there exists a user-item interaction history from which we aim to model user preferences over items. In particular, we consider a set of U users and a set of I items, where we use $u \in \{1, ..., U\}$ to index users and $i \in \{1, ..., I\}$ to index items. We consider the click[2] matrix $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_U]^\top \in \mathbb{N}^{U \times I}$ given by the user-item interactions. Its row vector $\mathbf{x}_u = [x_{u1}, ..., x_{uI}]^\top \in \mathbb{N}^I$ for $u \in \{1, ..., U\}$ denotes sample click counts over items for user u. We denote N_u as the total number of clicks from user u, i.e., $N_u = \sum_{i=1}^{I} x_{ui}$.

2.2 Related Work

Matrix factorization [12] has been a popular technique to solve the aforementioned problem in Section 2.1 [1, 3, 6, 16], factorizing the matrix of ratings into two classes of latent feature for user and items and, moreover, predicting missing or future rating via product of user and item factors. However, these matrix factorization based methods depend on the linearity assumption of the user-item interaction and cannot capture more complex, non-linear relationships between users and items. [13] showed that incorporating non-linear features to a hidden linear factor model can effectively improve the performances of the recommender systems.

With the expressive powers of the deep generative models, variational autoencoder (VAE) [10, 18] based methods have been proposed for collaborative filtering [9, 14, 20, 24, 25]. [14] proposed a VAE model which takes user-item scores as an input and learns a compressed late representations. The latent factors are then used to reconstruct the input scores and to compute the missing scores. We defer the more detailed description of the method in [14] to Section 2.3, serving as a building block for our proposed method.

2.3 Variational Autoencoder for Collaborative Filtering

VAEs are a class of probabilistic generative models using neural networks. VAEs are typically used to compress (encode) the input information into a multivariate latent distribution and to reconstruct (decode) the input as accurately as possible. Utilizing VAEs for collaborative filtering allows *non-linear* probabilistic latent-variable models, hence generalizing linear latent factor models, such as matrix factorization. The generative process of the proposed model in [14], Multi-VAE, follows similar procedure as the deep latent Gaussian model in [18]. For each user $u \in \{1, ..., U\}$, a latent variable $\mathbf{z}_u \in \mathbb{R}^K$ is sampled from a prior distribution. This sampled latent variable $\mathbf{z}_u$ is then fed into a generative model $f(\cdot)$ to compute the probability distribution of user's clicks over the items. In [14], a standard multivariate Gaussian distribution is used for the prior:

$$\mathbf{z}_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)\,,$$
$$\pi(\mathbf{z}_u) \propto \exp\{f(\mathbf{z}_u)\}\,,$$

where $\pi(\mathbf{z}_u) \in \mathbb{R}^I$ is a distribution over the I items for user u. Here, $f(\cdot)$ can be any differentiable function, where a linear function would reduce to classical matrix factorization. Therefore, this is a strict extension of the matrix factorization based latent variable models. In [14], a multilayer perceptron is used for $f(\cdot)$. Now, once $\pi(\mathbf{z}_u)$ is obtained, sampling of clicks $\mathbf{x}_u$ can be done. In particular, a multinomial distribution is assumed for the click distribution:

$$\mathbf{x}_u \sim \text{Multi}\left(N_u, \pi(\mathbf{z}_u)\right)\,. \tag{1}$$

Recall N_u is the total number of clicks by user u. $\mathbf{x}_u$ is generated using $\pi(\mathbf{z}_u)$ repeatedly over N_u trials. With this generative process, the VAE framework is then applied where the objective is to maximize the data likelihood $P(\mathbf{X}) = \int p(\mathbf{X}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$. Such an optimization problem, however, is challenging since the marginal likelihood of the data $P(\mathbf{X})$ is intractable under the function p_θ parametrized by θ. Addressing this challenge with variation inference, a lower-bound for the log marginal likelihood of the data is considered. Then, the objective becomes maximizing the lower-bound called the evidence lower bound (ELBO).

$$\begin{aligned}\log p(\mathbf{x}_u, \theta) &\geq \mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{x}_u)}\left[\log p_\theta(\mathbf{x}_u|\mathbf{z}_u)\right] - \text{KL}\big(q_\phi(\mathbf{z}_u|\mathbf{x}_u)\|p(\mathbf{z}_u)\big)\\ &:= \mathcal{L}(\mathbf{x}_u; \theta, \phi)\,.\end{aligned}$$

Here, $q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)$ is the variational distribution whose optimized function approximates the intractable posterior $p(\mathbf{z}_u \mid \mathbf{x}_u)$.

[2]Following the convention used in [14], we use the term "click" to represent any type of item consumption by a user, including "watch", "purchase", or "listen"

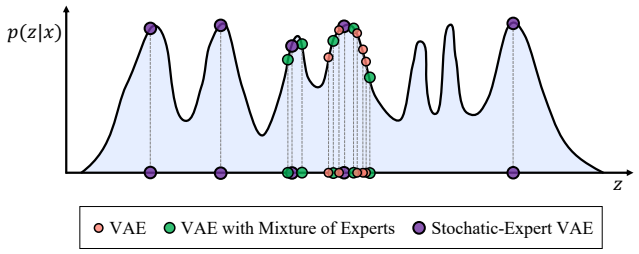

Figure 1: Conventional VAE approaches for collaborative filtering are not suitable for learning complex posterior distributions. While using a mixture distribution can help allow more flexible distributions, generative models based on mixture distributions may not be able to capture all the multiple modes when aggregated over multiple distributions via mixture. Our proposed *stochastic-exert* framework can overcome such a bottleneck via random selection of expert over multiple experts which allows sampling from different modes.

2.4 Challenges of VAE for Collaborative Filtering

One of the main challenges in the VAEs for collaborative filtering is that the prior (and as a result also the posterior) distributions may be too simple and restrictive to learn potentially rich and complex latent representation of user preferences. Various approaches [9] have been proposed to handle this issue with attempts to allow more flexible distributions. VAE with a mixture distribution [23] has been adapted to collaborative filtering in [9]. However, as we show later in the experiments, utilizing a mixture distribution alone does not appear to properly capture the richness of latent representation, resulting in similar performances of the VAEs without the mixture (e.g., `Multi-VAE` in [14]). See Section 5 for comparisons in the experimental results. One possible explanation for this observation is as follows. With a mixture distribution, there is a smoothing effect over multiple distributions which may be more expressive than a simple unimodal distribution. Yet, with a highly complex posterior distribution to be learned, it may only generate samples from the smoothed basins which may fail to capture various multiple modes. This phenomenon is illustrated in Figure 1.

In this work, we aim to overcome this issue via random sampling in expert selection instead of mixture aggregation. Sampling an expert for each item click is inspired by the generative process of Latent Dirichlet Allocation (LDA), where each word in a document is sampled from a selected topic from the document's topic distribution. We argue that this sampling expert scheme allows the model to effectively capture the multi-facet of users' preference.

3 PROPOSED METHOD

3.1 Model Description

In this section, we illustrate our proposed VAE model for collaborative filtering, which we call the *Stochastic-Expert Variational Autoencoder* or `SE-VAE` for short. As the naming suggests, our model considers a set of multiple experts, where each expert is associated with its own deep latent Gaussian model. What differentiates our method from the existing methods using multiple experts is that, instead of utilizing the multiple experts as a mixture aggregation, we incorporate *stochasticity in expert selection*. For clear and concrete exposition of our proposed framework, we describe the generative process of `SE-VAE` compared to `Multi-VAE` [14] explained in Section 2.3. It is important to note that our proposed methodology is generic and adaptable to any VAE achitecture, not confined to a specific network structure of `Multi-VAE`.

Multiple Experts. While the generative process of `Multi-VAE` starts by sampling a single latent variable $\mathbf{z}_u \in \mathbb{R}^K$ for user u from a standard Gaussian prior, our model maintains a set of M latent representations $\{\mathbf{z}_u^{(m)}\}_{m=1}^M$, where each representation corresponds to an expert – hence, we maintain a total of M experts. For user u and for expert m, the latent representation $z_u^{(m)}$ is transformed via a function $f_{\theta_m} : \mathbb{R}^K \to \mathbb{R}^I$ parametrized by θ_m to produce a probability distribution over I items, $\pi(\mathbf{z}_u^{(m)})$. That is, for each user $u \in \{1, ..., U\}$ and its expert $m \in \{1, ..., M\}$,

$$\mathbf{z}_u^{(m)} \sim \mathcal{N}(0, \mathbf{I}_K)\,,$$
$$\pi(\mathbf{z}_u^{(m)}) \propto \exp\{f_{\theta_m}(\mathbf{z}_u^{(m)})\}\,.$$

Here, the exponentiation of $f_{\theta_m}(\mathbf{z}_u^{(m)})$ is applied element-wise, $\exp(\mathbf{v}) = [\exp(v_1), ..., \exp(v_d)]$ for $\mathbf{v} \in \mathbb{R}^d$. For each expert m, $f_{\theta_m}(\cdot)$ is a multilayer perceptron, allowing each expert to keep its own parameter. The distribution over I items of expert m, $\pi(\mathbf{z}_u^{(m)})$, is obtained for every expert, and is used when generating *clicks*.

Stochastic Expert Selection. As mentioned earlier, the expert selection procedure is the key feature of our method. We first sample $\mathbf{w}_u \in \mathbb{R}^M$, the logits for the Gumbel-Softmax, which is used to select among multiple experts. Then, expert selection is sampled independently for each click $n \in \{1, ..., N_u\}$ for user u:

$$\mathbf{w}_u \sim \mathcal{N}(0, \mathbf{I}_M)\,,$$
$$\mathbf{e}_u^n \sim \text{Gumbel-Softmax}(\mathbf{w}_u)\,.$$

where the one-hot vector $\mathbf{e}_u^n \in \{0, 1\}^M$ represents an expert selection for the n-th click of user u. Hence, even for a single user u, clicks can be generated from different experts. Let m_n be the index of the selected expert for the n-th click, such that $\mathbf{e}_{u,m_n}^n = 1$. Then, a one-hot vector for a single click $\mathbf{x}_u^n$ is sampled from the distribution using the selected expert m_n:

$$\mathbf{x}_u^n \sim \text{Mult}\big(1, \pi(\mathbf{z}_u^{(m_n)})\big)\,.$$

Hence, the selected expert determines the distribution over the items, and therefore, an item-click is generated at the n-th click. In contrast to click sampling at the user level used in (1) for `Multi-VAE`, we can incorporate more flexible preferences by allowing multiple experts to generate samples for a single user. This procedure of an individual click sample is repeated for all $n \in \{1, ..., N_u\}$, hence creating a set of click vectors $\{\mathbf{x}_u^1, ..., \mathbf{x}_u^{N_u}\}$. Note that since click samples are independent of each other, one can generate a set of click vectors in parallel, rather than in a sequence. After generating each of the clicks, we aggregate over the individual click vectors to produce a click count vector for the given user, $\mathbf{x}_u$, i.e., $\mathbf{x}_u =$

$\sum_{n=1}^{N_u} \mathbf{x}_u^n$. This stochastic expert selection significantly differs from the previous work [14] and its variants, e.g., [9], where each user has its own *fixed* distribution over the items.

3.2 Variational Inference

Given the collection of the observed *clicks*, we are interested in estimating the parameters θ_m in the generative process shown in Section 3.1. As in the VAE, and its variants, we use variational inference to approximate the posterior $p(\mathbf{z}_u^{(m)} \mid \mathbf{x}_u)$ by a computationally tractable variational distribution. The variational parameters in the variational distribution are selected to minimize the Kullback-Leibler divergence between the true (and potentially intractable) posterior and the variational distribution. We use a factorized variational distribution over the latent variables $\{\mu_u^{(m)}, \sigma_u^{2,(m)}\}$ for each expert, this is consistent with the existing vanilla-VAE models with a fully factorized Gaussian distribution, which we extend by maintaining the parameters for each expert. Besides the set of $\{(\mu_u^{(1)}, \sigma_u^{2,(1)}), ..., (\mu_u^{(M)}, \sigma_u^{2,(M)})\}$, we again set $q(\mathbf{w}_u)$ to be a fully factorized Gaussian distribution which is used in Gumbel-Softmax as input logits. This weight logit vector allows us to maintain multiple experts and to sample an expert from the categorical distribution using the Gumbel-Softmax. The use of $\mathbf{w}_u$ sampled from a Gaussian distribution is motivated by the logistic normal used in correlated topic model [11]. With these variational parameters, the objective of variational inference is to minimize the Kullback-Leibler divergence by optimizing the free variational parameters.

3.3 Variational Auto-Encoder Implementation

We first provide a brief overview of the main three advantages that VAEs have over the conventional mean-field variational Bayes (VB) algorithms for variational inference. First, in VAEs, one can efficiently deal with intractable posterior distributions. Therefore, no simplified assumptions on the posterior distribution are required in VAEs. Second, VAEs can make parameter updates using small mini-batches or even single datapoints.[3] Third, the network parameters in the inference model (or the encoder) are shared across all data points allowing flexible reuse of inferences to answer related new problems, which is referred to as *amortized* inference [2, 14].

A VAE [10] consists of an encoder and a decoder. The encoder tries to discover some latent representation of an input in a probabilistic manner; while the decoder attempts to reconstruct the original input from the latent vector. In contrast to the conventional variational inference methods, where free variational parameters are updated independently, VAEs rely on neural network structure with data-dependent function. In the encoder of a VAE, a set of parameters denoted as ϕ are optimized to best learn the variational parameters for each user u: the K-dimensional mean and covariance vectors[4] with $g_\phi(\mathbf{x}_u) \stackrel{\text{def}}{=} [\mu_\phi(\mathbf{x}_u), \sigma_\phi(\mathbf{x}_u)]$. We extend this approach to ours by projecting M of $\{\mu_\phi(\mathbf{x}_u), \sigma_\phi(\mathbf{x}_u)\}$ through the encoder, having the inference function augmented as below:

$$g_\phi(\mathbf{x}_u) \stackrel{\text{def}}{=} \begin{bmatrix} \mu_\phi^{(1)}(\mathbf{x}_u) & \dots & \mu_\phi^{(M)}(\mathbf{x}_u) \\ \sigma_\phi^{(1)}(\mathbf{x}_u) & \dots & \sigma_\phi^{(M)}(\mathbf{x}_u) \end{bmatrix} \in \mathbb{R}^{2K \times M}, \tag{2}$$

where the $\mu_\phi(\cdot)$ and $\sigma_\phi(\cdot)$ are the variational parameters for K-dimensional mean and variance respectively as in the previous VAE models [10, 14]. For the (m)-th expert, the variational distribution is defined as follows:

$$q_\phi^{(m)}(\mathbf{z}_u^{(m)} \mid \mathbf{x}_u) = \mathcal{N}(\mu_\phi^{(m)}(\mathbf{x}_u), \text{diag}\{\sigma_\phi^{2,(m)}(\mathbf{x}_u)\}). \tag{3}$$

which reflects how the encoder with parameter ϕ outputs the corresponding variational parameters of the variational distribution from the input data $\mathbf{x}_u$. From Equation 2, it is worth noting that the parameters in the encoder are shared across all of the experts rather than having separate encoders for each. Later in our experiments, we discuss how the former approach outperforms the latter approach.

As mentioned earlier, our proposed model consists of multiple experts, where one of the experts is stochastically selected for each *click* in the generative process. Therefore, we use an additional neural network for the Gumbel-Softmax distribution. This additional network is then connected to our bottleneck layer of `SE-VAE` acting as a channel selector. During the training phase, the Gumbel-Softmax layer outputs a one-hot vector and otherwise generates the soft mixtures.

The overall generative model with the encoder and the decoder is summarized in Figure 2. The output from the Gumbel-Softmax layer act as a channel selector. When an expert is selected, the encoded input data is processed through the selected decoder. Later, each reconstruction is aggregated for computing the reconstruction loss. In the following, we provide the details.

Evidence Lower Bound. For the inference, we resort to variational inference in the VAE context by starting with the *lower bound* of the marginal log-likelihood, namely Evidence Lower Bound (ELBO). In a collaborative filtering context, users are assumed to be independent of each other, and the marginal log-likelihood becomes the sum over the marginal log-likelihood of each user's data point. The ELBO of the marginal log-likelihood for user u is defined as $\mathcal{L}(\mathbf{x}_u; \theta, \phi)$, where for simplicity of expression, we denote the collection of parameters $\{\theta^{(1)}, ..., \theta^{(M)}\}$ as θ:

$$\begin{aligned} \log p(\mathbf{x}_u; \theta) &\geq \mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{x}_u)}[\log p_\theta(\mathbf{x}_u|\mathbf{z}_u)] \\ &\quad - \text{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)||p(\mathbf{z}_u)) - \text{KL}(q_\phi(\mathbf{w}_u \mid \mathbf{x}_u)||p(\mathbf{w}_u)) \\ &\stackrel{\text{def}}{=} \mathcal{L}(\mathbf{x}_u; \theta, \phi). \end{aligned} \tag{4}$$

Note that the second component in the RHS of the KL-divergence accounts for the logits which is fed into the Gumbel-Softmax layer. Taking the stochastic expert in to account, the ELBO can be further expressed as:

$$\begin{aligned} \mathcal{L}(\mathbf{x}_u; \theta, \phi) = &\sum_{m=1}^{M} \mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{x}_u)}[q_{\text{cat}}(\mathbf{e}_{u,m} = 1) \log p_{\theta_m}(\mathbf{x}_u|\mathbf{z}_u^{(m)})] \\ &- \text{KL}(q_\phi(\mathbf{z}_u \mid \mathbf{x}_u)||p(\mathbf{z}_u)) - \text{KL}(q_\phi(\mathbf{w}_u \mid \mathbf{x}_u)||p(\mathbf{w}_u)), \end{aligned} \tag{5}$$

where $q_{\text{cat}}(\mathbf{e}_{u,m} = 1)$ is a probability of expert (m) being selected for user u. The training procedure involves maximizing the ELBO

[3] In the strict sense, a recent work, Stochastic-Variational Inference [5] applies stochastic optimization to scale up recent advances in variational inference.

[4] Note that we use a diagonal covariance matrix, hence only requiring a K-dimensional vector for covariance.

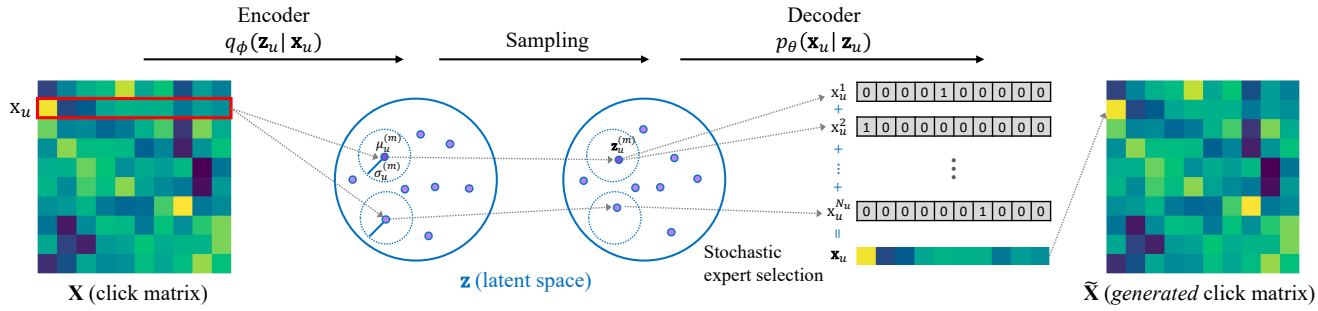

Figure 2: Schematic overview of SE-VAE. The click data $\{\mathbf{x}_u\}_u$ gets encoded via an encoder network into latent representation $\{\mathbf{z}_u^{(m)}\}_{u,m}$. Each one-hot encoded click vector $\mathbf{x}_u^n$ for $n \in \{1, ..., N_u\}$ is then generated using randomly selected expert via a decoder network to re-construct a click count vector $\mathbf{x}_u$ for user u.

through updating ϕ and θ. The ELBO can also be interpreted as the expected negative reconstruction error with a regularizer. The first term of the Equation 5 is a negative reconstruction error in auto-encoder parlance; while the second term acts as a regularizer controlling KL-divergence of the approximate posterior from the prior. It is well known that the KL-divergence can be integrated analytically by having the prior as $p(\mathbf{z}) = \mathcal{N}(0, I)$ and the posterior approximation $q_\phi(\mathbf{z}|\mathbf{x})$ as Gaussian of which the subscript and superscript have been omitted. In our study, we follow the same assumptions made in VAE [10] and previous work [9, 14] in collaborative filtering. Due to the stochasticity in the sampling process, performing back-propagation becomes challenging. Here, we utilize *reparametrization trick* referring to the original work VAE [10] and categorical reparametrization [8] for Gaussian and Gumbel-Softmax sampling respectively. The overall learning process is provided in Algorithm 1.

Algorithm 1 Training procedure for SE-VAE

Require: $\mathbf{X} \in \mathbb{N}^{U \times I}$
Ensure: $\phi, \theta = \{\theta^{(1)}, ..., \theta^{(M)}\}$
  Initialize θ, ϕ
  while not converged **do**
    Obtain batch of users
    for user u in a batch **do**
      Sample $\mathbf{z}_u$ and $\mathbf{w}_u$ using the RT
      Compute gradient of $\mathcal{L}$ w.r.t. θ, ϕ with $\mathbf{z}_u$ and $\mathbf{w}_u$
    end for
    Take average of gradients from the batch
    Update θ and ϕ with SGD
  end while

β-VAE [4] is a modification of the VAE framework. It has been shown that the β-VAE is more stable and achieves better performance than the vanilla-VAE which is when $\beta = 1$ in β-VAE. If β is small, then the influence of the prior constraint are weakened. While having β set to $\beta > 1$ [4] yields improvements in performance in visual domain. Liang *et al.* [14] found that setting β set to $\beta < 1$ is more effective in collaborative filtering. The findings in later studies [9, 20] are also consistent with [14]. As in [9, 14, 20], The KL-divergence in the ELBO for our model has β set to $\beta < 1$ for our study.

4 EXPERIMENTAL SETUP

Before presenting our experimental results in Section 5, we first describe the experimental setup for the empirical evaluations of our proposed method, SE-VAE, and the benchmark models. We then provide the descriptions of the benchmark datasets and the evaluation metrics widely used for evaluating recommender systems.

4.1 Model

In our experiments, we compare the performance of our model against benchmark models, which is presented in Section 5. We set one of the best performing models from [9] as our benchmark, where we follow the exact same settings from their experiments. As such, throughout our experiments we keep K the latent space dimension to 200 the have the number of hidden units in each layer fixed to 600, which are also consistent with Multi-VAE [14] and its following work by Kim and Suh [9]. For a fair comparison, possible parameters were fixed to the same settings in [9] including batch sizes, and learning rates. We use the same configurations for other settings, such as split of training, validation, and test datasets as done in [9] for its predecessors.

For evaluating our model, we considered SE-VAE with three experts. As we show in the experimental results, this number of experts appears to be a suitable choice, exhibiting superior performances over the existing methods although the number is relatively small and not optimized for performances. We believe the fine-tuning on the number of experts would enhance the performances of the proposed method, which we leave for future work. For the logistic normal distribution used for sampling experts, we set its prior to the same number for each expert. The model was implemented and trained using PyTorch. The experiments were conducted using a GeForce RTX 2080 Ti GPU. Code is avaiable at https://github.com/yoonsikcho/se-vae.

4.2 Datasets

We evaluate our proposed algorithm using the datasets which have been widely used for collaborative filtering. The datasets used in

this study is in the same format as in previous studies [9, 14, 15, 20, 21, 25], which has been binarized reflecting implicit feedback. The explicit feedback rating has been binarized having a threshold of 3, meaning the rating 4 and above out of 5 has been marked as 1, otherwise 0. Inactive users with less than 5 reviews have been removed from each dataset.

MovieLense20M. This is a user-movie rating dataset. Each user left their rating score on a scale of 5, which has been binarized for our study. After preprocessing the dataset, MovieLense20M dataset contains 136, 677 users with 20, 108 items. Between these users and the given items, there were 10.0 million interactions. On average, a user has clicked 73 items, and the % of interactions is 0.36%.

Netflix. This is a user-movie rating dataset from the Netflix Prize. The rating scale is the same as the MovieLense20M. The same preprocessing method has been taken in this dataset. After preprocessing the dataset, the Netflix dataset contains 435, 435 users nearly 3.4 times the users from MovieLense20M. The dataset contains 17, 769 items with 56.9 million interactions. On average, a user has clicked 122 items, and the % of interactions is 0.69%. Netflix data has denser input data $\mathbf{X}$ than that of MovieLense20M.

4.3 Evaluation Metrics

We follow the same metrics which have been used in previous studies [9, 14, 15, 20, 21, 25]. Two ranking-based metrics: Recall@R and the truncated Normalized Discounted Cumulative Gain (NDCG@R) are used throughout our experiments. These two metrics both compare the predicted ranking of the held-out items (in validation and testing) with their true ranking, where the rankings prediction can be obtained from the output of the decoder $p_\theta(\cdot)$ based on the inferred latent representation. We summarize both metrics below:

Recall@R. In collaborative filtering, we are interested in predicting the top-N items to the user in a sense that top-N items are more carefully observed in recommender systems. Recall@R is the proportion of relevant (clicked) items predicted in the top R items. This metric becomes useful considering that the online users focus on top-N recommended items on first page or upper area without scrolling down.

NDCG@R. Recall@R considers all items equally important when predicted within the first R. This becomes problematic when R becomes large as it cannot differentiate the importance between items when they are all in top-R. Normalized discounted cumulative gain overcomes this issue by using the monotonically increasing discount. It emphasizes the importance of higher ranking than lower ones.

5 RESULTS

Here we discuss the empirical performances of our proposed method and comparison with the existing methods on the main evaluation task, *click* prediction. For our experiments, we use the dataset and evaluation metrics summarized in Sections 4.2 and 4.3 respectively. The purposes of the evaluations are three-fold:

(1) We want to show that our proposed method outperforms the existing state-of-the-art methods.
(2) We want to illustrate the flexibility of our framework, easily adapting to various VAE models used in CF.
(3) We want to compare our approach with other possible model extensions including models with a higher value of K for the latent dimension, and a mixture of experts.

5.1 Performance Comparison against the Benchmark Models

For our first experiment, we apply the stochastic expert (SE) framework to the current state-of-the-art models in [9], the Hierarchical VampPrior with Gated Linear Units (GLU), `H+VAMP(Gated)`, and compare its performance to that of the previous benchmark models [7, 9, 14, 15, 17, 20, 21, 24, 25]. For the implementation of SE, we used three experts. The experiment results show the superior performances of `H+VAMP(Gated)` with SE from the state-of-the-art, even only with three experts. As the results are shown in Table 1, our proposed model in the bottom row outperforms the previous benchmark models in every aspect. The best performing model and the best results in each metric are marked in bold.

For the results shown in Table 1, the same test datasets from `Mult-VAE` [14] were used for all the models including our method. The results for `WMF` [7], `SLIM` [17], and `CDAE` [24] are from the experiment reports in [14], where the performance of each model was evaluated based on the same test datasets. All the other results are taken from the respective original papers. We find that both `VAEGAN` [25] and `EASE` [21] follow the same preprocessing procedure as `Mult-VAE`. `VAEGAN` and `EASE` use `Mult-VAE` as their baselines citing the results of `Mult-VAE` in [14]. We confirm that the codes for `RaCT` [15], `RecVAE` [20], and `H+Vamp(Gated)` [9] also use the same test datasets as those of `Mult-VAE`. Our implementation is reformulated based on `H+Vamp(Gated)` for fair comparisons, with evaluations using the exact same test datasets.

5.2 Flexibility of SE-VAE

Now, we further investigate whether the performance improvement can be shown for the SE adaptation to other VAE models. In our second experiment, we verify the flexibility of `SE-VAE` by applying the SE framework to various VAE models, which includes `Mult-VAE` and the enhanced `Mult-VAE` with two hidden layers and GLU from [9]. As shown in Table 2, we observe that the VAE models incorporating the SE framework consistently outperform their corresponding VAE models without the SE, achieving higher performance in every metric. This result suggests that the simple modification of adapting the SE to the existing methods provide performance improvement, hence allowing for wide applicability of the proposed framework.

5.3 Ablation Studies

In the following experiments, we validate the effectiveness of our model by comparing `SE-VAE` against a VAE with a bottleneck layer with a higher dimension, matching the number of variational parameters of ours, and with other possible models with a similar idea. In the following, we discuss this in detail.

Bottleneck Layer Dimension. As elaborated previously, the output layer of the encoder in `SE-VAE` has more units, M-times the number of the VAE. In our implementation, we distribute the sampled $\mu_u^{(m)}$

Table 1: Comparing our proposed method against the benchmark models. Our proposed method (SE-VAE): a model in which stochastic-expert (SE) is applied to the previous state-of-the-art model, H+VAMP(Gated). The results for WMF [7], SLIM [17], CDAE [24] are from the experiment reports in Multi-VAE [14], all the other results are taken from the respective original papers. The standard errors of the performances across the models are around 0.002 for MovieLens20M and 0.001 for Netflix.

Model	MovieLens20M			Netflix		
	NDCG@100	Recall@50	Recall@20	NDCG@100	Recall@50	Recall@20
WMF [7]	0.386	0.498	0.360	0.351	0.404	0.316
SLIM [17]	0.401	0.495	0.370	0.379	0.428	0.347
CDAE [24]	0.418	0.523	0.391	0.376	0.428	0.343
Mult-VAE [14]	0.426	0.537	0.395	0.386	0.444	0.351
VAEGAN (AVB+D+C) [25]	0.438	0.541	0.407	0.396	0.447	0.363
EASE [21]	0.420	0.521	0.319	0.393	0.445	0.362
RaCT [15]	0.434	0.543	0.403	0.392	0.450	0.357
RecVAE [20]	0.442	0.553	0.414	0.394	0.452	0.361
H+Vamp (Gated) [9]	0.445	0.551	0.413	0.408	0.462	0.376
SE-VAE (H+Vamp, Gated)	**0.447**	**0.556**	**0.418**	**0.409**	**0.463**	**0.377**

Table 2: Comparing SE-VAE models with the VAE benchmark models. For benchmark models, results are taken from [9]. Denoted SE in the model name when stochastic expert has been applied. Stochastic expert has been applied to each model in [9]. Standard errors are around 0.002 for MovieLens20M and 0.001 for Netflix.

Model	MovieLens20M			Netflix		
	NDCG@100	Recall@50	Recall@20	NDCG@100	Recall@50	Recall@20
Mult-VAE	0.42700	0.53524	0.39569	0.38711	0.44427	0.35255
Mult-VAE (**SE**)	**0.43057**	**0.53688**	**0.40010**	**0.38789**	**0.44512**	**0.35332**
Mult-VAE (Gated)	0.43515	0.54498	0.40558	0.39241	0.44958	0.35953
Mult-VAE (Gated,**SE**)	**0.44163**	**0.54594**	**0.41229**	**0.39342**	**0.45094**	**0.36010**
H+Vamp (Gated)	0.44522	0.55109	0.41308	0.40861	0.46252	0.37678
H+Vamp (Gated,**SE**)	**0.44718**	**0.55551**	**0.41787**	**0.40907**	**0.46312**	**0.37713**

and $\sigma_u^{(m)}$ to its expert (m) accordingly. Figure 3 presents the overall structure of the SE-VAE, where the connected Gumbel-Softmax layer and its network (Figure 1.b.-right) has been omitted for simplicity. To further verify that our performance is not simply due to the increased number of units in the bottleneck layer, we perform another set of experiments. In this small experiments, we use the MovieLens20M data, and compare the NDCG@100 obtained from the benchmark models with increased number of units with the results of ours. Table 3 verifies the performance gain through SE is not merely due to the higher number of neurons. Left column ($K = 200$) is taken from [9], center column (K=600) is the results obtained when the total number of units in the bottleneck layer matches that of our model. Right column is taken from our results.

Table 3: Comparison between our proposed algorithm with baseline models. When the latent dimension is set to $K = 600$ the number of parameters become same as our proposed algorithm with 3 experts. Results in NDCG@100 are reported using the MovieLens20M dataset.

Models (NDCG@100)	$K = 200$	$K = 600$	with SE
Mult-VAE	0.42700	0.42709	**0.43057**
Mult-VAE (Gated)	0.43515	0.42482	**0.44163**
H+Vamp (Gated)	0.44522	0.44485	**0.44718**

Mixture of Expert vs Stochastic Expert. In this set of experiments, we verify whether the stochastic expert is effective by comparing with another widely known VAE extension: the mixture of experts (MoE). We use MovieLense20M data to compare the performances of the two extended models. For the three models selected from [9], we apply MoE and SE on each of the three models and report their performances. Table 4 reveals that SE-VAE outperforms the MoE extensions. Interestingly, when MoE has been applied to VAE, the performance always drop from the original VAE models. The mixture assumption is not strong enough to capture the complex user preference. We believe the performance drop of MoE could be due to the mixing of different modalities, where each effectiveness washes away contrary to our expectations.

Sharing an Encoder vs Individual Encoder per Expert. Our proposed model shares the parameter ϕ as shown in Figure 3. A natural

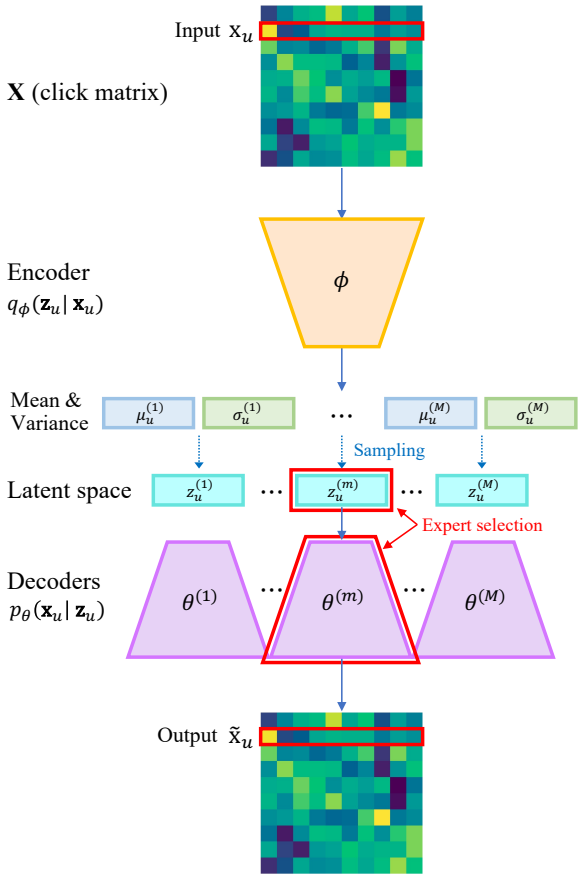

Figure 3: Network structure of SE-VAE. For visual simplicity, we depict only the VAE assuming one-hot expert is sampled from other network. The encoder is shared across all the experts, while the decoders are assigned for each of the expert. For each of the click, one-hot expert vector selects a decoder and make predictions through $p_\theta(\cdot)$.

Table 4: Comparison between mixture of experts and stochastic expert in their NDCG@100. Applying stochastic expert on each model outperforms mixture of experts approach, while mixture of experts approach drops the original performance. MovieLens20M dataset has been used for this report.

Models (NDCG@100)	original	with MoE	with SE
Mult-VAE	0.42700	0.42601	**0.43057**
Mult-VAE (Gated)	0.43515	0.43216	**0.44163**
H+Vamp (Gated)	0.44522	0.43142	**0.44718**

question that arises from the proposed model can be the the following: "What if the encoder is assigned to each expert? " To answer this question, we also conducted experiments using the MovieLense20M dataset, where we compared a SE model with separate encoders with our proposed approach. We found that the model with individual encoder showed the performance similar to that of single VAE. We hypothesize that the model with individual encoder exhibits this behavior because this approach is, in fact, similar to averaging the results from multiple running of single VAE. On the other hand, when we use a shared encoder, when updating the variational parameters $\{\mu^{(m)}, \phi^{(m)}\}_{m=1}^{M}$, the update of a variational parameter pair (μ, σ) affects the other pairs. We believe this better reflects the multifaceted nature of human activities.

6 CONCLUSION

In this paper, we propose a novel framework of VAE for collaborative filtering on implicit feedback data. The proposed framework incorporates a new feature where individual experts are sampled stochastically at each user-item interaction. Our model is able to effectively utilize the variability across multiple experts, and the stochasticity in expert selection provides performances gains which are evidenced by the comprehensive numerical experiments. Based on the experiments with real-world benchmark datasets from MovieLens and Netflix, our proposed method showed superior performances compared to the existing state-of-the-art collaborative filtering methods across all metrics considered. We further provide additional experiments comparing the proposed stochastic expert framework with other mixture approaches, which reveals the strength of our model design.

While we focus on the problem of collaborative filtering in this paper, we believe that the proposed *stochastic expert* technique can be used to enhance VAEs in general beyond the application of collaborative filtering. Hence, this novel technique can be of independent interest. In our future work, we plan to further explore the effectiveness of the stochastic expert technique in general VAEs.

ACKNOWLEDGMENTS

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2021R1F1A1063389) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-01341, AI Graduate School Program Chung-Ang University, No.2021-0-02067, Next Generation AI for Multi-purpose Video Search).

REFERENCES

[1] Daniel Billsus, Michael J Pazzani, et al. 1998. Learning collaborative information filters.. In *Icml*, Vol. 98. 46–54.
[2] Samuel J. Gershman and Noah D. Goodman. 2014. Amortized Inference in Probabilistic Reasoning. *Cognitive Science* 36 (2014).
[3] Prem Gopalan, Jake M Hofman, and David M Blei. 2015. Scalable Recommendation with Hierarchical Poisson Factorization.. In *UAI*. 326–335.
[4] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.
[5] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic Variational Inference. *Journal of Machine Learning Research* 14, 4 (2013), 1303–1347. http://jmlr.org/papers/v14/hoffman13a.html
[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.
[7] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. 263–272. https://doi.org/10.1109/ICDM.2008.22

[8] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumbel-Softmax. In *Proceedings International Conference on Learning Representations 2017*. OpenReviews.net. https://openreview.net/pdf?id=rkE3y85ee

[9] Daeryong Kim and Bongwon Suh. 2019. Enhancing VAEs for Collaborative Filtering: Flexible Priors & Gating Mechanisms. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 403–407. https://doi.org/10.1145/3298689.3347015

[10] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[11] John Lafferty and David Blei. 2006. Correlated Topic Models. In *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt (Eds.), Vol. 18. MIT Press. https://proceedings.neurips.cc/paper/2005/file/9e82757e9a1c12cb710ad680db11f6f1-Paper.pdf

[12] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788–791.

[13] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM conference on recommender systems*. 59–66.

[14] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 689–698. https://doi.org/10.1145/3178876.3186150

[15] Sam Lobel, Chunyuan Li, Jianfeng Gao, and Lawrence Carin. 2020. RaCT: Toward Amortized Ranking-Critical Training For Collaborative Filtering. In *International Conference on Learning Representations*.

[16] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[17] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE Computer Society, USA, 497–506. https://doi.org/10.1109/ICDM.2011.134

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*. PMLR, 1278–1286.

[19] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[20] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (Houston, TX, USA) *(WSDM '20)*. Association for Computing Machinery, New York, NY, USA, 528–536. https://doi.org/10.1145/3336191.3371831

[21] Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*. 3251–3257.

[22] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[23] Jakub Tomczak and Max Welling. 2018. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1214–1223.

[24] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 153–162. https://doi.org/10.1145/2835776.2835837

[25] Xianwen Yu, Xiaoning Zhang, Yang Cao, and Min Xia. 2019. VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4206–4212. https://doi.org/10.24963/ijcai.2019/584