Towards Deeper Understanding of Variational Auto-encoders for Binary Collaborative Filtering

Siamak Zamani
Cognitive Computing Lab
Baidu Research
10900 NE 8th St
Bellevue, WA 98004, USA
zamanys4@gmail.com

Dingcheng Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St
Bellevue, WA 98004, USA
dingchengl@gmail.com

Hongliang Fei
Cognitive Computing Lab
Baidu Research
10900 NE 8th St
Bellevue, WA 98004, USA
feihongliang0@gmail.com

Ping Li
Cognitive Computing Lab
Baidu Research
10900 NE 8th St
Bellevue, WA 98004, USA
pingli98@gmail.com

ABSTRACT

Recommendation systems are an integral component of machine learning, wherein collaborative filtering (CF) is among the most prominent algorithms employed. Recently, variational auto-encoders (VAEs) with multinomial likelihood and weighted Kullback-Leibler (KL) regularization (referred to as Mult-VAE) provide state-of-the-art performance for collaborative filtering of binary data. To gain deeper insight into the objective function of Mult-VAE, we build a connection between the reconstruction term of Mult-VAE objective and the objective function of the probabilistic n-Choose-k model for ranking prediction. In particular, we theoretically demonstrate that the negative reconstruction error of Mult-VAE is a lower bound to the log-likelihood of the binary n-Choose-k model. Hence, Mult-VAE can be interpreted as an approximate proxy to the n-Choose-k model. We also empirically show the essential role of this reconstruction term of evidence lower bound in the context of collaborative filtering on multiple real-world datasets. Finally, inspired by the role of the weighted KL term in maximizing mutual information between observed ratings and latent variables, we propose a semi-implicit VAE framework with superior performance in terms of ranking metrics.

CCS CONCEPTS

• **Computing methodologies** → *Learning latent representations*; • **Information systems** → **Collaborative filtering**; • **Theory of computation** → *Models of learning*.

KEYWORDS

Recommendation systems, Variational auto-encoders, collaborative filtering, deep generative models, n-Choose-k model

ACM Reference Format:
Siamak Zamani, Dingcheng Li, Hongliang Fei, and Ping Li. 2022. Towards Deeper Understanding of Variational Auto-encoders for Binary Collaborative Filtering. In *ACM International Conference on the Theory of Information Retrieval, July 14–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539813.3545145

1 INTRODUCTION

Recommender systems have been widely adopted by many online services, including E-commerce [44], streaming services [15], and social media sites [36]. The goal of a recommendation system is to show users a set of previously unseen items that are likely to be of interest to them. Collaborative filtering (CF) is among the most prominent algorithms employed in recommendation systems [18]. he basic ideas of CF have been integrated into massive-scale click-through rate (CTR) prediction deep learning models in commercial search engines [9, 11, 52, 59]. CF methods predict user preferences by discovering similarity patterns across users and items.

Many CF methods rely on latent factor models for prediction of user-item interactions [21, 24, 41]. As an example, methods based on *matrix factorization* are among some of the most successful realizations of latent factor models for CF [22, 23, 30**?**]. The linear nature of such methods, however, may restrict their modeling capacity, therefore a growing body of work applies neural networks in the CF context [17, 50, 55].

Recently, variational auto-encoders (VAEs) [29, 39, 51] have been used to produce state-of-the-art results in the CF setting [34]. The objective function of VAE, known as evidence lower bound (ELBO), consists of two terms; reconstruction error or distortion term, and the KL-regularization term. In particular, VAEs are attractive for settings with large-scale datasets, as they amortize the inference by employing encoder networks. In practice, many user-item interactions are only inferred implicitly, and thus encoded as binary matrices. Liang et al. [34] develop a VAE based method with multinomial likelihood, Mult-VAE, and achieve state-of-the-art performance on several real-world binary CF datasets. In addition to employing multinomial likelihood, another essential step in making Mult-VAE [34] perform well in the CF problem is KL annealing, where the weight of the Kullback-Leibler (KL) regularization in the objective function of VAE is adjusted based on the validation data to allow for learning more informative latent representations. In fact, this is a well known problem of the VAE, referred to as *latent variable collapse* [7], where powerful likelihood models lead to good generative models, while lacking useful latent representations [2, 58].

In this paper, we shed light on the use of multinomial likelihood by establishing a connection between the objective function of Mult-VAE and that of a probabilistic model for ranking, known as n-Choose-k model [46]. More precisely, n-Choose-k model is a probabilistic framework developed for multi-class recognition and ordinal regression problems. Swersky et al. [46] show that optimal decision theoretic predictions under the n-Choose-k model for

---

ICTIR'22, July 14–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-9412-3/22/07…$15.00
https://doi.org/10.1145/3539813.3545145

monotonic gain functions such as normalized discounted cumulative gain (NDCG) can be achieved by a simple sorting operation.

We demonstrate both theoretically and empirically that the negative reconstruction error of Mult-VAE [34] is a lower bound to the log-likelihood of the binary n-Choose-k model. Hence, Mult-VAE can be interpreted as an approximate proxy to the n-Choose-k model [46], whose optimization in large-scale settings is intractable. This view is further approved, as our empirical experiments show that discarding the KL term completely leads only to a slight decrease in the performance of Mult-VAE, and equivalently the dominating role of the reconstruction term in achieving a good performance in ranking prediction.

Finally, to improve the performance of VAE for collaborative filtering, we proposed a new extension based on semi-implicit VAE framework [53]. More specifically, we use a reparameterizable implicit distribution as a mixing distribution to effectively expand the richness of the variational family. In particular, semi-implicit VAE employs a hierarchical encoder that injects random noise at different stochastic layers. This added noise, with skip connections to input data in the encoder architecture, leads to increasing the mutual information between latent variables and observed data, and thus better performance in predicting the missing ratings.

Overall, the contributions of our work are as follows:

- We show that the negative reconstruction term in the evidence lower bound (ELBO) of Mult-VAE is a lower bound approximation for the log-likelihood of the binary n-Choose-k model, and thus attribute the reconstruction term as a key factor in determining the performance of Mult-VAE for collaborative filtering.
- We empirically demonstrate the essential role of the reconstruction term on several real-world datasets.
- We propose a semi-implicit VAE based model to improve the performance in collaborative filtering.

The remainder of this paper is as follows. Section 2 introduces neural network based collaborative filtering models and variational auto-encoders. Section 3 first provides a brief review of VAE and Mult-VAE for collaborative filtering, then presents the main theoretical contribution of our work. Experimental studies verifying the theories are demonstrated in section 4, in which we also discuss an additional direction, parameterized dropouts, as another potential way for improving the performance of VAEs for collaborative filtering. We conclude the paper with the main insights in section 5.

2 RELATED WORKS

Before neural network based models prevailing, matrix factorization [22, 23, 31] is the most widely used method for Collaborative filtering. It uses an inner product of the user-item matrix to quantify user-item interactions. Some model specifications further include global user/item biases and/or regularization terms to prevent overfitting. An advanced extension is the use of Gaussian Markov Random Field (MRF) to model the dependency among items [45], which leveraged sparse inverse covariance estimation with autoencoders and neighborhood models. Since our focus is on deep generative model based CF, we do not cover them in details.

Different from matrix factorization based approaches, neural Collaborative Filtering(NCF) replaces the user-item inner product with a neural architecture. Neural-network-based collaborative filtering models started by focusing on explicit feedback data [13, 42, 43, 54, 56], and gradually shifted attention to implicit feedback data [34, 56]. For instance, collaborative denoising auto-encoder (CDAE) [50] augments the standard denoising auto-encoder by adding a per-user latent factor to the input. A disadvantage of CDAE, compared to VAE, is that the number of its model parameters grows linearly with the number of users as well as items, making it more prone to over-fitting. Furthermore, in the testing phase, CDAE requires additional optimization to learn the latent representation of unseen users. Neural collaborative filtering (NCF) [17] presents a model with non-linear interactions between the user and item latent factors rather than the commonly used dot product. Similar to CDAE, the number of parameters of NCF grows linearly with both the number of users and items, and thus making it problematic to apply NCF to large datasets.

Meanwhile, variational auto-encoders [29, 39, 51] are widely applied to image [8] and text data [35]. Despite being able to capture complex distributions, vanilla variational auto-encoders underperform their counterparts when used for collaborative filtering. This can be associated with *latent variable collapse* phenomenon, i.e., the variational posterior is independent of data [7]. This collapse happens as minimizing the KL term in the ELBO leads to a degenerate solution, where $q_\phi(z|x) \approx p(z)$, x is the input, z is the latent variable and ϕ are encoder parameters.

Many recent works based on VAEs have attempted to alleviate this problem by maximizing the mutual information between observed and latent variables [1, 7, 20, 57]. In particular, as reweighting the KL term in the ELBO with small weights is associated with maximizing the mutual information between observations and latent representations [58], [34] were first to use this intuition to improve the performance of VAE for collaborative filtering. As it has been only recently to be used in the context of collaborative filtering, to the best of our knowledge this paper is the only work pointing out the relationship between VAE and the probabilistic n-Choose-k model. Meanwhile, several efforts integrated VAEs with other well-known neural models to better fit in the collaborative setting. For example, aWAE [56] extended the Wasserstein autoencoders [47] in the collaborative filtering problem to tack the overlapping issues in the distributions of latent variables of the encoder. VAEGAN [54] combined VAE and Generative Adversarial Network (GAN) to better approximation to the posterior. However, as illustrated in Rosca et al. [40], such VAE-GAN hybrid models are harder to scale, evaluate, and use for inference compared to VAEs.

3 METHOD

In this section, we first present a brief background on VAEs, then describe how they are successfully applied in the context of collaborative filtering. The connection between the reconstruction term of VAE with multinomial likelihood and the binary n-Choose-k model is then discussed in detail. Motivated by the insights from these analysis, we introduce our proposed semi-implicit VAE for collaborative filtering.

3.1 Variational Auto-Encoder

Let $\boldsymbol{x} \in \mathbb{R}^I$ be a vector of I observable variables and $\boldsymbol{z} \in \mathbb{R}^K$ a vector of stochastic latent variables, where usually in practice we have $K << I$, to obtain useful low-dimensional latent representations. A variational auto-encoder (VAE) [29, 39] binds together modeling and inference, where the model is a parametric joint distribution of observed and latent variables,

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}),$$

where $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is the generative model (the *decoder*), typically constructed using a neural network with parameters θ, and $p(\boldsymbol{z})$ is the *prior* distribution on the latent representations.

Given observed data $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, the model parameters θ are learned by maximizing the marginal likelihood of the observations,

$$\begin{aligned} \theta^* &= \arg\max_\theta \; \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right] \\ &= \arg\max_\theta \; \frac{1}{N}\sum_{i=1}^{N}\log\int p_\theta(\boldsymbol{x}_i, \boldsymbol{z}_i)d\boldsymbol{z}_i, \end{aligned}$$

where $p_{\mathcal{D}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}_i)$ is the empirical distribution of the observed data. Performing this optimization is difficult in practice, as each term of the objective function contains an integral that is intractable most of the time. To overcome this issue, VAEs rely on amortized variational inference to approximate the optimization. More precisely, the evidence lower bound (ELBO) is used as a surrogate objective function,

$$\begin{aligned} \mathcal{L}(\theta, \phi) &:= \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}\Big[\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big]\Big] \\ &\leq \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x})\right], \end{aligned} \tag{1}$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is the data-dependent variational posterior (the *encoder*), usually constructed using a neural network with parameters ϕ. The set of model parameters $\{\theta, \phi\}$ are then jointly learned using stochastic gradient optimization methods [29, 39].

3.2 VAE for Collaborative Filtering

In the context of binary collaborative filtering, data is in the form of user-by-item interaction matrix $\boldsymbol{X} \in \{0, 1\}^{U\times I}$, where U and I are number of users and items, respectively. In this work, we consider collaborative filtering with implicit feedback [24]. Specifically, the entry x_{ui} of the interaction matrix equals one, when user u has interacted with item i, for example the user has clicked on a content related to the item, and zero when no interaction has been observed.

Liang et al. [34] propose Mult-VAE, an extension of the VAE framework with multinomial likelihood for collaborative filtering with implicit feedback. Specifically, for each user u, the generative process of Mult-VAE starts by sampling a K-dimensional latent representation $\boldsymbol{z}_u$ from a standard multivariate Gaussian prior,

$$\boldsymbol{z}_u \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_K).$$

The latent representation $\boldsymbol{z}_u$ is then transformed via a non-linear function, e.g. a multi-layer perceptron (MLP) combined with a softmax function, to produce a probability distribution $\pi(\boldsymbol{z}_u)$ over different items. Finally, the observed interactions for user u, $\boldsymbol{x}_u$, are generated using a multinomial likelihood,

$$\boldsymbol{x}_u \sim \text{Mult}\big(N_u, \pi(\boldsymbol{z}_u)\big), \tag{2}$$

where N_u is the number of items which user u has interaction with.

To learn the latent representation of users in Mult-VAE, similar to Gaussian VAE [29, 39], data-dependent variational distributions over latent variables $\boldsymbol{z}_u$ are employed,

$$q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u) = \mathcal{N}\big(\mu_\phi(\boldsymbol{x}_u), \text{diag}\{\sigma^2_\phi(\boldsymbol{x}_u)\}\big), \tag{3}$$

where $\mu_\phi(\boldsymbol{x}_u)$ and $\sigma_\phi(\boldsymbol{x}_u)$ are non-linear transformations of the observed data, e.g. achieved by a MLP block.

To learn the model parameters $\{\theta, \phi\}$, a natural step in variational inference is to maximize the ELBO in (1), which can also be expressed as,

$$\begin{aligned} \mathcal{L}(\theta, \phi) &= \mathbb{E}_{\boldsymbol{x}_u\sim p_{\mathcal{D}}(\boldsymbol{x}_u)}\Big[\mathbb{E}_{\boldsymbol{z}_u\sim q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)}\left[\log p_\theta(\boldsymbol{x}_u|\boldsymbol{z}_u)\right] \\ &\quad - \text{KL}\big(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)||p(\boldsymbol{z}_u)\big)\Big], \end{aligned} \tag{4}$$

where KL is the Kullback-Leibler divergence.

Optimizing ELBO for collaborative filtering, however, can prove inefficient in practice as the KL term in (4) can be minimized by simply letting variational posteriors and priors on latent representations be equal, and thus preventing the VAE from learning useful data-dependent latent variables [2]. To overcome this issue, similar to β-VAE [**?**], [34] consider an alternative objective function to ELBO with weighted KL term,

$$\begin{aligned} \mathcal{L}_\beta(\theta, \phi) &= \mathbb{E}_{\boldsymbol{x}_u\sim p_{\mathcal{D}}(\boldsymbol{x}_u)}\Big[\mathbb{E}_{\boldsymbol{z}_u\sim q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)}\left[\log p_\theta(\boldsymbol{x}_u|\boldsymbol{z}_u)\right] \\ &\quad - \beta\cdot\text{KL}\big(q_\phi(\boldsymbol{z}_u|\boldsymbol{x}_u)||p(\boldsymbol{z}_u)\big)\Big], \end{aligned} \tag{5}$$

with $\beta < 1$. In fact, it can be shown that using weights $\beta < 1$ leads to maximization of the mutual information between latent and observed variables [58]. To find the best value for β in the context of collaborative filtering, [34] perform a heuristic search, where they start training with $\beta = 0$ and gradually increase β to 1. They record the best β when the performance of Mult-VAE in terms of a ranking metric such as NDCG on a validation dataset reaches the peak.

3.3 Binary Collaborative Filtering

Despite using multinomial likelihood in the generative model (2), Mult-VAE has been primarily applied to binary observed variables, obtaining state-of-the-art performance results for large scale datasets [34]. In this section, we provide a theoretical analysis on why using a multinomial objective function for binary collaborative filtering leads to optimization of monotonic ranking gains such as NDCG [25] which is the standard measure in learning to rank for search engines [33]. We further show similar theoretical properties for Bernoulli likelihoods.

We start by expanding the first term in (5), which can be viewed as the *negative reconstruction error* or *negative distortion* [2]. In the rest discussions, the index for user u is dropped for simplicity. For a single data sample $\boldsymbol{x}$, the reconstruction term can be expressed as,

$$\mathbb{E}_{z\sim q_\phi(z|\boldsymbol{x})}\big[\log p_\theta(\boldsymbol{x}|z)\big] \propto \mathbb{E}_{z\sim q_\phi(z|\boldsymbol{x})}\Big[\sum_{i=1}^{I} x_i \log \pi_i(z)\Big]$$

$$= \sum_{i=1}^{I} \mathbb{E}_{z\sim q_\phi(z|\boldsymbol{x})}\Big[x_i\Big(f_{\theta,i}(z) - \log\big(\sum_{i'}\exp\{f_{\theta,i'}(z)\}\big)\Big)\Big], \qquad (6)$$

where $f_{\theta,i}$ is the output of the decoder MLP, corresponding to item i. The expectations in (6) are intractable. Hence, to calculate the gradient of the negative reconstruction term in (6), *reparameterization* trick for gradient estimation is employed [29, 39]. More precisely, the variational posterior (3) can be equivalently expressed as $z_\phi = \mu_\phi(\boldsymbol{x}) + \sigma_\phi(\boldsymbol{x})\boldsymbol{\epsilon}$, where the random variable $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}_K)$ does not depend on encoder parameters ϕ. Thus the expectations in (6) can be re-written as expectations with respect to $\boldsymbol{\epsilon}$

$$\mathbb{E}_{z\sim q_\phi(z|\boldsymbol{x})}\Big[x_i\Big(f_{\theta,i}(z) - \log\big(\textstyle\sum_{i'}\exp\{f_{\theta,i'}(z)\}\big)\Big)\Big]$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I}_K)}\Big[x_i\Big(f_{\theta,i}(z_\phi) - \log\big(\textstyle\sum_{i'}\exp\{f_{\theta,i'}(z_\phi)\}\big)\Big)\Big]. \qquad (7)$$

Using a single Monte Carlo sampling can yield low variance approximation to the expectation in (7) [5], thereby the final form of negative distortion term used in Mult-VAE can be expressed as,

$$\sum_{i=1}^{I}\Big\{x_i\Big(\eta_i - \log\big(\sum_{i'}\exp\{\eta_{i'}\}\big)\Big)\Big\}, \qquad (8)$$

where we have introduced $\eta_i := f_{\theta,i}(z_{\boldsymbol{\phi}})$ with implicit dependency on θ and ϕ.

3.4 Connection to Binary n-Choose-k Model

We now establish a connection between the form of the reconstruction term in (8) and the objective function of the Binary n-Choose-k model of [46].

Assuming model inputs $\boldsymbol{\eta} = (\eta_1, ..., \eta_I)$ and binary outputs $\boldsymbol{x} = (x_1, ..., x_I)$ with N_u elements equal to 1, the generative process of the Binary n-Choose-k model is as follows:

- Draw N_u from a prior distribution over counts.
- Draw a subset $\boldsymbol{c} \subset \{1, ..., I\}$ with cardinality N_u to have values 1, according to the following probability:

$$p(\boldsymbol{x_c} = \boldsymbol{1}, \boldsymbol{x_{\bar{c}}} = \boldsymbol{0}) = \frac{\exp\{\sum_{i\in\boldsymbol{c}}\eta_i\}}{\sum_{\boldsymbol{x}|\sum_i x_i = N_u}\exp\{\sum_i x_i\eta_i\}}.$$

Swersky et al. [46] show that the maximization of the likelihood in probabilistic n-Choose-k model leads to the optimal decision-theoretic prediction for monotonic ranking gain functions such as NDCG. We provide the main theorem from [46] for completeness of presentation.

Theorem 1. *Under an ordinal n-Choose-k model, the optimal decision theoretic predictions for a monotonic ranking gain are made by sorting* $\boldsymbol{\eta}$ *values.*

Given N_u, to learn the model parameters using maximum likelihood, one should optimize

$$\log p(\boldsymbol{x}; \boldsymbol{\eta}) = \sum_{i=1}^{I} x_i\eta_i - \log\Big(\sum_{\boldsymbol{x}|\sum_i x_i = N_u}\exp\{\sum_i x_i\eta_i\}\Big). \qquad (9)$$

The similar forms of (8) and (9) suggest a connection between the reconstruction error of VAE with multinomial likelihood and the binary n-Choose-k model. The following theorem shows that in fact the negative reconstruction term in (8) is a lower bound of (9).

Theorem 2. *Given* $N_u \geq 1$ *and model inputs* $\boldsymbol{\eta}$*, the negative reconstruction term in (8) is a lower bound to the log-likelihood of binary n-Choose-k model.*

Proof. The second term of the negative reconstruction term can be written as

$$N_u \log\big(\sum_{i'}\exp\{\eta_{i'}\}\big) = \log\Big(\sum_{i'}\exp\{\eta_{i'}\}\Big)^{N_u}.$$

We then expand the argument of logarithm using multinomial expansion:

$$\Big(\sum_{i'}\exp\{\eta_{i'}\}\Big)^{N_u}$$

$$= \sum_{n_1+...+n_I=N_u}\binom{N_u}{n_1, ..., n_I}\exp\big\{\sum_{i'} n_{i'}\eta_{i'}\big\}$$

$$\geq \sum_{n_1+...+n_I=N_u}\exp\big\{\sum_{i'} n_{i'}\eta_{i'}\big\}$$

$$\geq \sum_{x_1+...+x_I=N_u}\exp\{\sum_{i'} x_{i'}\eta_{i'}\},$$

where we use the facts that for $N_u \geq 1$, $\binom{N_u}{n_1,...,n_I} \geq 1$ and $\{\boldsymbol{x}|x_1 + ... + x_I = N_u\}$ for binary x_i is a subset of $\{\boldsymbol{n}|n_1 + ... + n_I = N_u\}$ for integer n_i. $\square$

Theorem 2 sheds new light on the reason that using a VAE with multinomial likelihood, as in Mult-VAE [34], can be beneficial for binary collaborative filtering. Although reconstruction term is only a part of the objective function of Mult-VAE in (5), we observe that in practice values of $\beta << 1$ lead to peak performance in terms of ranking accuracy, and thereby the negative reconstruction term dominates the objective function. In fact, our experimental results in the following section show that dropping the KL term in (5) completely, and thus optimizing only the reconstruction term, results in minimal performance deterioration, consistently across different datasets.

In addition to a VAE with multinomial likelihood, we demonstrate that a similar lower bound to the log-likelihood of binary n-Choose-k model can be obtained by employing a Bernoulli-logistic combination instead of multinomial-softmax of Mult-VAE. Specifically, the negative reconstruction term under this framework after applying re-parameterization trick and monte carlo approximation can be expressed as

$$\sum_{i=1}^{I}\Big\{x_i\eta_i - \log\big(1 + \exp\{\eta_i\}\big)\Big\}. \qquad (10)$$

The following theorem shows that the negative reconstruction term of VAE with Bernoulli-logistic likelihood is a lower-bound to the log-likelihood of the binary n-Choose-k model.

Theorem 3. *Given $N_u \geq 1$ and model inputs $\boldsymbol{\eta}$, the negative reconstruction term in (10) is a lower bound to the log-likelihood of binary n-Choose-k model.*

Proof. Similar to the proof of theorem 2, we show that the second term in (10) is a lower-bound of the second term of (9). We have,

$$\begin{aligned}\exp\Big\{\sum_i \log\big(1+\exp\{\eta_i\}\big)\Big\} &= \prod_i \big(1+\exp\{\eta_i\}\big)\\ &= \sum_{x_1+\ldots+x_I\leq I} \exp\{\sum_{i'} x_{i'}\eta_{i'}\}\\ &\geq \sum_{x_1+\ldots+x_I=N_u} \exp\{\sum_{i'} x_{i'}\eta_{i'}\},\end{aligned}$$

where x_i is binary. Taking logarithms of the both sides of the above inequality we reach the desired result. □

In addition to multinomial and Bernoulli distributions, we can consider the more general exponential family of distributions as the likelihood of the decoder network. For any distribution in this family, we have

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \nu(\boldsymbol{x})\exp\big\{\boldsymbol{\eta}^T\boldsymbol{x} - A(\boldsymbol{\eta})\big\},$$

where $\boldsymbol{\eta}$ is the natural parameter and $A(\cdot)$ is the log-normalizer of the family. Hence, the negative distortion function corresponding to a likelihood from exponential family can be written as $\sum_{i=1}^{I}\big\{x_i\eta_i - A(\eta_i)\big\}$. Therefore, if $\sum_i A(\eta_i)$ is an upper-bound of $\sum_{x_1+\ldots+x_I=N_u}\exp\{\sum_i x_i\eta_i\}$, then the corresponding negative reconstruction term can serve as a proxy of the log-likelihood of n-Choose-k model.

3.5 KL Annealing

From an information-theoretic perspective, maximizing the modified lower-bound in (5) with $\beta < 1$ amounts to maximizing the mutual information between latent and observed variables. More precisely, we can rewrite (5) as [58]:

$$\begin{aligned}-\mathcal{L}_\beta(\theta,\phi) &= (\beta-1)I_{q_\phi}(\boldsymbol{x};\boldsymbol{z}) + \beta\mathrm{KL}(q_\phi(\boldsymbol{z})||p(\boldsymbol{z}))\\ &+ \mathbb{E}_{q_\phi(\boldsymbol{z})}\big[\mathrm{KL}(q_\phi(\boldsymbol{x}|\boldsymbol{z})||p_\theta(\boldsymbol{x}|\boldsymbol{z}))\big] \qquad (11)\end{aligned}$$

where $I_{q_\phi}(\boldsymbol{x};\boldsymbol{z}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})p_{\mathcal{D}}(\boldsymbol{x})}\big[\log\frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{q_\phi(\boldsymbol{z})}\big]$ is the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ under the variational distribution, and $q_\phi(\boldsymbol{z}) = \int q_\phi(\boldsymbol{z}|\boldsymbol{x})p_{\mathcal{D}}(\boldsymbol{x})d\boldsymbol{x}$ is the *aggregate posterior*. Hence, optimizing $\mathcal{L}_\beta(\theta,\phi)$ with values of β smaller than one lead to increments in the mutual information between observed and latent variables as a side product.

Another relationship between reconstruction term, KL term and the mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ is presented in [2] as $H - D \leq I_{q_\phi}(\boldsymbol{x};\boldsymbol{z}) \leq R$, where $H = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}\big[-\log p_{\mathcal{D}}(\boldsymbol{x})\big]$ corresponds to the *entropy* of the underlying data source, D is the reconstruction error, and $R = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}}(\boldsymbol{x})}\big[\mathrm{KL}\big(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\big)\big]$ is the *rate* function. Thus, minimizing the distortion function (reconstruction term) can be interpreted as maximizing the lower bound of the mutual information without setting any explicit constraints on the rate function.

The bounds in this section and the previous one suggest that to achieve good performance in collaborative filtering, we need to optimize the mutual information or maximize the lower bound for n-Choose-k model. Indeed, our empirical studies show that modifying the objective function without taking these relationships into account, such as adding weight decaying terms can worsen the performance of Mult-VAE as they violate the aforementioned lower bound properties.

3.6 Semi-Implicit VAE for Collaborative Filtering

Equipped with insights from previous sections, we understand why Mult-VAE cannot perform better no matter how we tune parameters in experiments. Nonetheless, Equation (11) sheds light on how to further improve the performance of collaborative filtering under the VAE framework. Namely, if a variant of VAE can optimize the mutual information, the lower bound of the n-Choose-K-model can be further maximized to approach the ELBO. It is found that semi-implicit Variational inference (SIVI) [53] has the desirable properties to meet this goal, as illustrated below. Therefore, we now propose to use SIVI to enhance the performance of VAE for collaborative filtering.

In its general form, SIVI defines the approximate posterior in a hierarchical manner as:

$$\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{\psi}), \quad \boldsymbol{\psi} \sim q_\phi(\boldsymbol{\psi}),$$

where $\boldsymbol{\psi}$ is the mixing distribution. Its marginalization leads to the Variational family $\mathcal{H} = \{h_\phi(\boldsymbol{z}) : h_\phi(\boldsymbol{z}) = \int_{\boldsymbol{\psi}} q(\boldsymbol{z}|\boldsymbol{\psi})q_\phi(\boldsymbol{\psi})d\boldsymbol{\psi}\}$.

To obtain expressive Variational posterior distributions, the mixing distribution $q_\phi(\boldsymbol{\psi})$ is allowed to be implicit (e.g. transformation of noise through an MLP), thus the marginal $h_\phi(\boldsymbol{z})$ is intractable and for estimating model parameters the following lower bound to ELBO is optimized [53]:

$$\begin{aligned}\mathcal{L}_K(\phi,\theta) &= \mathbb{E}_{\boldsymbol{\psi}\sim q_\phi(\boldsymbol{\psi})}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{\psi})}\mathbb{E}_{\boldsymbol{\psi}^{(1)},\ldots,\boldsymbol{\psi}^{(K)}\sim q_\phi(\boldsymbol{\psi})}\Big[\\ &\log\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{\frac{1}{K+1}\big[q(\boldsymbol{z}|\boldsymbol{\psi}) + \sum_{k=1}^{K} q(\boldsymbol{z}|\boldsymbol{\psi}^{(k)})\big]}\Big],\end{aligned}$$

where as the number of samples K increases, the above lower bound approaches ELBO.

In an amortized setting, rather than using a single-stochastic-layer encoder as in VAE, semi-implicit VAE (SIVAE) can add as many stochastic layers as needed, as long as the first stochastic layer is reparameterizable and has an analytic PDF, and the layers added after are reparameterizable and simple to sample from. More specifically, we use the following hierarchical construction with random noise injection at M layers:

$$\begin{aligned}q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\mu},\Sigma) &= \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}),\Sigma_\phi(\boldsymbol{x})),\\ \boldsymbol{\mu}_\phi(\boldsymbol{x}) = f_\phi(\boldsymbol{\ell}_M,\boldsymbol{x}), &\quad \Sigma_\phi(\boldsymbol{x}) = g_\phi(\boldsymbol{\ell}_M,\boldsymbol{x}),\\ \boldsymbol{\ell}_t = T_t(\boldsymbol{\ell}_{t-1},\boldsymbol{\epsilon}_t,\boldsymbol{x};\phi), &\quad \boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t) \text{ for } t=1,\ldots,M \qquad (12)\end{aligned}$$

where $\boldsymbol{\ell}_0 = \emptyset$ and f, g and T_t are all deterministic neural networks. Note that in each layer of the encoder, in addition to the output of previous layer and the random noise, the observed inputs are also injected. This construction is closely related to skip connections, which are shown to increase the mutual information between observed and latent random variables [7]. To learn the model parameters, we leverage stochastic gradient descent as outlined in Algorithm 1.

Algorithm 1: Semi-Implicit VAE for collaborative filtering.

```
Input: Data $\{\boldsymbol{x}_u\}$, neural networks $T_t$, $f_\phi$ and $g_\phi$, source of
       randomness $\boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t)$, step sizes $\rho$ and $\delta$
Output: Model parameters $\phi$ and $\theta$
Initialize $\phi$ and $\theta$ randomly
$\boldsymbol{\psi} = (\boldsymbol{\mu}_\phi, \boldsymbol{\mu}_\Sigma)$
while not converged do
    Set $\mathcal{L}_K = 0$
    Sample $\boldsymbol{\psi}^{(k)}$ for $k = 1, ..., K$ as in (12)
    Take sub-sample $\{\boldsymbol{x}_u\}_{u_1:u_M}$
    for $j = 1 : J$ do
        Sample $\boldsymbol{\psi}_j$ and $\boldsymbol{z}_j$ according to (12)
        $\mathcal{L}_K = \mathcal{L}_K + \frac{1}{J}\Big[ -\log \frac{1}{K+1}\big[ \sum_{k=1}^{K} q(\boldsymbol{z}_j|\boldsymbol{\psi}^{(k)}) +$
          $q(\boldsymbol{z}_j|\boldsymbol{\psi}_j)\big] + \frac{N}{M}\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_j) + \log p(\boldsymbol{z}_j)\Big]$
    end
    $\phi = \phi + \rho\nabla_\phi\mathcal{L}_K$
    $\theta = \theta + \delta\nabla_\theta\mathcal{L}_K$
end
```

4 EXPERIMENTS

We first study the impact of discarding the KL regularization term in the VAE framework, i.e. training the models only using the distortion function term. We consider collaborative filtering using both multinomial (8) and Bernoulli (10) likelihoods. We then study how common techniques for improvement of VAEs may affect or not affect the performance in binary collaborative filtering problem.

Throughout the experiments, we follow [34] and split all users into training, validation and test sets. The model is trained using the entire interactions in the training set. For validation and testing, 80% of the interactions are randomly chosen to learn the user-level latent representations, and then the ranking evaluation metrics are calculated based on the model predictions of the 20% held-out interactions.

For both encoder and decoder networks we use MLPs with one hidden layer with width 600. We set the dimension K of latent variable $\boldsymbol{z}$ to 200. Our experiments show that using deeper architecture does not enhance the performance. We use $tanh(\cdot)$ activation function for all MLPs. To tune the regularization parameter β in (5), we follow [34] and anneal the KL term linearly for 200,000 gradient updates. For training, we use Adam optimizer [27] with batch size of 500.

4.1 Datasets

We use three widely used large-scale user-item interaction datasets in our empirical studies as below.

MovieLens-20M (ML-20M)[1]: This is a user-movie ratings dataset gathered from a movie recommendation service [16]. Users who have watched less than five movies are discarded. The user-movie matrix is binarized by keeping ratings of four or higher.

Table 1: Attributes of datasets used in the experiments. Interactions are non-zero entries of the user-item matrix.

	ML-20M	Netflix	MSD
# of users	136,677	463,435	571,355
# of items	20,108	17,769	41,140
# of interactions	10.0M	56.9M	33.6M
# of held-out users	10,000	40,000	50,000

Netflix Prize (Netflix)[2]: This movie-user ratings data is from the Netflix Prize [3]. We perform similar pre-processing steps as for ML-20M, and only keep users watched at least five movies, and binarize the data by keeping ratings of four or higher.

Million Song Dataset (MSD)[3]: This is a user-song play counts data released as part of the Million Song Dataset [4]. We follow the same procedure as in [34] to binarize the data: we only keep users with at least 20 songs in their listening history and songs that are listened to by at least 200 users.

Following Liang et al. [34], we have the same training/validation/test set split. Table 1 contains the attributes of the three datasets. % of interactions means the density of the user-item click matrix. # of the held-out users refers to the number of validation/test users out of the total number of users in the first row.

4.2 Metrics

We use two popular learning-to-rank scoring functions to compare the predicted rank of the held-out items with their true rank: truncated normalized discounted cumulative gain (NDCG@R) and Recall@R. Let ω be a permutation of R items, $\omega(r)$ be the item at rank r, and I_u be the set of held-out items that user u has interactions with. Then, we have

$$\begin{aligned} \text{Recall@}R(u,\omega) &:= \frac{\sum_{r=1}^{R} \mathbb{I}[\omega(r) \in I_u]}{\min(R, |I_u|)}, \\ \text{DCG@}R(u,\omega) &:= \sum_{r=1}^{R} \frac{2^{\mathbb{I}[\omega(r)\in I_u]} - 1}{\log(r+1)}. \end{aligned}$$

NDCG@R is then obtained by dividing DCG@R by its maximum possible value, where all the held-out items are ranked at the top. NDCG@R uses a monotonically increasing discount coefficient to emphasize the significance of higher ranks versus lower ones, while Recall@R considers all items equally important.

4.3 Baselines

In addition to Mult-VAE [34], we also include the performance results of the following state-of-the-art collaborative filtering models, with similar settings as in [34]:

Weighted Matrix Factorization (WMF) [24]: WMF is a linear low-rank factorization model, which is trained by alternating least squares. The weights on the 0's is set to one, and the 1's are tuned from the set $\{2, 5, 10, 30, 50, 100\}$. The dimension of latent variables

---

[1] https://grouplens.org/datasets/movielens/20m/

[2] https://archive.org/details/nf_prize_dataset.tar

[3] http://millionsongdataset.com/

is selected as $K \in \{100, 200\}$ based on NDCG@100 on validation users with a grid search method.

SLIM [37]: This is a sparse linear model which learns an item-to-item similarity matrix by solving a constrained ℓ_1-regularization optimization. The regularization parameters are found using a grid-search over $\{0.1, 0.5, 1, 5\}$. Due to long run-time, the results of SLIM on MSD dataset are not reported.

Collaborative Denoising Auto-encoder (CDAE) [50]: This model augments the standard denoising auto-encoder by adding per-user latent factors to the input. The dimension of the bottleneck layer is set to 200, and Adam optimizer [27] with weight decay is used for training.

aWAE [56]: This model introduced an L_1 regularization term to Wasserstein autoencoders to learn a sparse low-rank representation form for the latent variables of VAE. We directly adopt their reported results on ML-20M and Netflix benchmarks for comparison.

4.4 Removing the KL Term

In the first part of the experiments, we quantitatively assess how the VAE based collaborative filtering method performs if the KL regularization term is completely discarded, i.e. in (5) we have $\beta = 0$.

Table 2 contains the performance results of Mult-VAE [34], and also the scenarios where only the reconstruction term based on multinomial likelihood (8), and the reconstruction term based on Bernoulli likelihood (10) are used for training the encoder and decoder networks. These results support the implications of Theorems 2 and 3, which present the particular form of reconstruction terms obtained by multinomial-softmax and Bernoulli-logistic combinations as lower bound approximates of the n-Choose-k model, justifying the good performance of these variants in terms of monotonic ranking metrics. This observation de-emphasizes the role of KL annealing for achieving state-of-the-art performance in the collaborative filtering, as advocated for in [34]. We further notice that using the Bernoulli-logistic combination in the decoder network can lead to slightly better performances, when the KL annealing is not employed. This observation underscores the importance of the sub-optimality connections established in Theorems 2 and 3, and refute the claim of superiority of multinomial over Bernoulli outlined in [34].

Table 2: Comparison between Mult-VAE and the methods without KL regularization term, which are solely based on the distortion function. "Multinomial" and "Logistic", respectively, denote the distortion functions based on multinomial and Bernoulli likelihoods.

(a) ML-20M

	Recall@20	Recall@50	NDCG@100
Mult-VAE	0.395	0.537	0.426
Multinomial	0.386	0.530	0.418
Logistic	0.390	0.526	0.423

(b) Netflix

	Recall@20	Recall@50	NDCG@100
Mult-VAE	0.351	0.444	0.386
Multinomial	0.344	0.438	0.380
Logistic	0.352	0.438	0.386

(c) MSD

	Recall@20	Recall@50	NDCG@100
Mult-VAE	0.266	0.364	0.316
Multinomial	0.254	0.349	0.300
Logistic	0.266	0.353	0.315

4.5 Impacts of VAE variants

In this section of experiments, we study the impacts of adopting some of the commonly used approaches for the improvement of VAE on the performance of VAE based collaborative filtering. We divide these approaches into two groups, according to whether they alter the form of the reconstruction function or not. In the first category, we have the methods that are used to improve the performance of VAEs by employing more sophisticated distributions.

Variational Mixture of Posteriors (VampPrior) [48]. This framework extends VAE with *variational mixture of posteriors* prior, which consists of mixture distributions with components given by variational posteriors conditioned on learnable *pseudo-inputs*. Specifically, VampPrior can be expressed as

$$p_\lambda(z) = \frac{1}{M} \sum_{m=1}^{M} q_\phi(z|\boldsymbol{u}_m),$$

where $\boldsymbol{u}_m$ is a K-dimensional vector referred to as pseudo-inputs, M is the number of pseudo-inputs, and $\lambda = \{\phi, \boldsymbol{u}_1, ..., \boldsymbol{u}_M\}$ is the set of parameters for this prior. Note that the encoder network parameters ϕ are shared between the prior and variational posterior, under this framework. In our experiments we use $M = 5$.

Inverse Autoregressive Flow (IAF) [28]. This framework, which consists of a chain of invertible transformations, aims to build flexible variational posterior distributions. Each transformation is based on an autoregressive neural network such as MADE [14]. Briefly, the tth transformation with input $z^{(t-1)}$ and output $z^{(t)}$ is constructed as

$$\begin{aligned} \boldsymbol{\sigma}^{(t)} &= \text{sigmoid}(\boldsymbol{s}^{(t)}), \\ \boldsymbol{z}^{(t)} &= \boldsymbol{\sigma}^{(t)} \odot \boldsymbol{z}^{(t-1)} + (1 - \boldsymbol{\sigma}^{(t)}) \odot \boldsymbol{m}^{(t)}, \end{aligned}$$

where $[\boldsymbol{m}^{(t)}, \boldsymbol{s}^{(t)}]$ is the output of the autoregressive network with input $\boldsymbol{z}^{(t)}$. In our experiments, we use $T = 1$ layer of transformation, hence $\boldsymbol{z}^{(0)}$ corresponds to the original latent variable with diagonal covariance matrix in (3), and $\boldsymbol{z}^{(1)}$ is a latent variable with full covariance matrix.

Regularization of NN weights. Finally, we consider regularizing the weights of neural networks in both encoder and decoder. Specifically, the objective function with regularization can be expressed as

$$\mathcal{L}_\beta(\theta, \phi) + \lambda \mathcal{R}(\boldsymbol{w}),$$

Table 3: Comparisons of various baselines and different configurations of VAE with multinomial likelihood. VAE+Vamp: VampPrior is used as the prior on latent variables z, VAE+IAF: inverse autoregressive flow is used in the encoder network, and VAE+Reg: a regularization term on weights of neural networks is added to the objective function. Best results are in bold.

(a) ML-20M

	Recall@20	Recall@50	NDCG@100
WMF	0.360	0.498	0.386
SLIM	0.370	0.495	0.401
CDAE	0.391	0.523	0.418
aWAE	**0.400**	0.530	0.429
Mult-VAE	0.395	0.537	0.426
VAE+Vamp	0.395	0.538	0.426
VAE+IAF	0.387	0.525	0.416
VAE+Reg	0.328	0.458	0.360
SIVAE	**0.400**	**0.539**	**0.430**

(b) Netflix

	Recall@20	Recall@50	NDCG@100
WMF	0.316	0.404	0.351
SLIM	0.347	0.428	0.379
CDAE	0.343	0.428	0.376
aWAE	0.352	0.438	0.386
Mult-VAE	0.351	0.444	0.386
VAE+Vamp	0.352	0.443	0.386
VAE+IAF	0.340	0.430	0.375
VAE+Reg	0.275	0.359	0.311
SIVAE	**0.358**	**0.448**	**0.391**

(c) MSD

	Recall@20	Recall@50	NDCG@100
WMF	0.211	0.312	0.257
SLIM	-	-	-
CDAE	0.188	0.283	0.237
aWAE	-	-	-
Mult-VAE	0.266	0.364	0.316
VAE+Vamp	0260	0.357	0.310
VAE+IAF	0.247	0.343	0.298
VAE+Reg	0.090	0.140	0.120
SIVAE	**0.272**	**0.373**	**0.323**

where $\mathcal{R}(\cdot)$ is the regularizer and $\boldsymbol{w}$ is the set of weights. In our experiments, we use $\lambda = 0.01$ and a ℓ_2 regularization function.

Table 3 compares the performance results of SIVAE and different configurations of VAE with baseline methods in terms of ranking evaluation metrics. By constructing more flexible approximate posterior distributions, and also increasing the mutual information between observed and latent variables, SIVAE is able to improve the performance of Mult-VAE across all benchmark datasets.

More carefully examining Table 3, two major observations can be noted. First, both VampPrior and IAF variants fail to improve the performance of Mult-VAE for ranking the held-out items. In fact, employing IAF slightly decreases the performance. This observation can be justified based on the experiments in section 4.4, and also the connections established between the distortion function of Mult-VAE and the objective function of probabilistic n-Choose-k model. More precisely, the distortion function dominates the objective function when the goal is to maximize the ranking metrics, and as indicated by Theorems 1 and 2, by optimizing this reconstruction function one can approximately optimize the monotonic ranking functions such as NDCG. Employing VampPrior and IAF, however, does not change the general form of the distortion function in (8), as they only modify the way $\boldsymbol{\eta}$ depends on the inputs $\boldsymbol{x}$.

On the other hand, Table 3 shows that incorporating a regularization term for the weights of the neural networks severely deteriorates the performance of Mult-VAE for collaborative filtering. This may be justified by noting that introducing the regularization term breaks the lower-bound relationship of the distortion function of Mult-VAE to the objective of n-Choose-k model. Thus, increments in the objective of Mult-VAE with regularization do not necessarily translate into increments in (9).

These observations illustrate the limitations of general techniques for improvement of VAEs in achieving better ranking predictions.

4.6 Discussion

Given the importance of the reconstruction term for the performance of VAE in collaborative filtering, a natural question arising is how to exploit this vision to improve the performance in ranking predictions. In our work, we have exploited one solution, semi-implicit variational inference (SIVI) [53], to expand the commonly used analytic variational distribution family by mixing the variational parameter with a flexible distribution. In particular, semi-implicit VAE employs a hierarchical encoder that injects random noise at different stochastic layers. This noise injection is closely related to the dropout step, utilized in the input layer of Mult-VAE. In fact, our experimental observations in Table 3 confirm the utility of semi-implicit VAE, which is a type of inherent regularization on the reconstruction term, to gain state-of-the-art ranking prediction performance. Thereby, improving the dropout step, by introducing learnable parameters can be another potential direction to improve the optimization of the reconstruction term [12].

Witnessing the success of transformers [6, 26, 32, 49], it is natural to introduce a transformer-based VAE [10, 38] to overcome the limitations of VAE for collabrative filtering. As we know, a main reason for using VAE is to remedy the no-set-meaning issue of latent codes in a general auto-encoder. The regularising loss in VAE encourages a smooth distribution of latent codes. However, VAE for collabrative filtering, in practice, is extremely prone to overfitting as the network learns to place all the probability mass to the non-zero entries. It has been proved that the maximization of the likelihood in probabilistic n-Choose-k model leads to the optimal decision-theoretic prediction for monotonic ranking gain functions such as NDCG [46]. Therefore, how to maximize the likelihood becomes a key in this regard. Integration of transformer into VAE may create a program to meet the need since multi-head attentions will help the scoring or ranking of user-item relations follow the semantic relevance to a much larger degree. Namely, the representations of latent codes with transformer-based VAE will improve the loss against the target output. We can evaluate the latent code effectiveness and even use Bayesian optimization to search the space. This exploration is our future work.

5 CONCLUSION

In addition to the role of reweighted KL term in increasing the mutual information between learned representations and observed data, in this paper we established a theoretical connection between the distortion (reconstruction error) term of the evidence lower bound of Mult-VAE, which offers an explanation for usage of multinomial likelihood for binary collaborative filtering. We also derived a similar connection for VAE with Bernoulli likelihood. These connections suggest VAE as an amortized variational approximation of the probabilistic n-Choose-k model, which has theoretical guarantees for optimization of monotonic ranking metrics such as NDCG. Our empirical experiments show the major role of the distortion term in the good performance of Mult-VAE for ranking prediction task, thus reducing the importance of the KL annealing step. Based on our theoretical analysis, we also justify why employing techniques such as mixture priors or more sophisticated encoder networks may fail to result in better ranking predictions. This sheds more light on the limitations of VAE for collaborative filtering. We also proposed a framework based on semi-implicit variational inference to improve the performance of VAE in collaborative filtering. Evaluation on multiple real-world benchmarks demonstrates the utility of SIVAE in boosting the performance of VAE based collaborative filtering methods. Finally, we had a detailed and deep discussions on the implications of the findings and proposed future approaches to further improve the performances of current AVEs.

REFERENCES

[1] Alessandro Achille and Stefano Soatto. 2018. Information Dropout: Learning Optimal Representations Through Noisy Computation. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 12 (2018), 2897–2905.

[2] Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. 2018. Fixing a Broken ELBO. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Stockholmsmässan, Stockholm, Sweden, 159–168.

[3] James Bennett, Stan Lanning, et al. 2007. The netflix prize.

[4] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Miami, FL, 591–596.

[5] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. 2016. Importance Weighted Autoencoders. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. San Juan, Puerto Rico.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2019), 4171–4186.

[7] Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. 2019. Avoiding Latent Variable Collapse with Generative Skip Models. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. Naha, Okinawa, Japan, 2397–2405.

[8] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

[9] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. MOBIUS: Towards the Next Generation of Query-Ad Matching in Baidu's Sponsored Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. Anchorage, AK, 2509–2517.

[10] Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828* (2021).

[11] Hongliang Fei, Jingyuan Zhang, Xingxuan Zhou, Junhao Zhao, Xinyang Qi, and Ping Li. 2021. GemNN: Gating-enhanced Multi-task Neural Networks with Feature Interaction Learning for CTR Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. Virtual Event, Canada, 2166–2171.

[12] Yarin Gal, Jiri Hron, and Alex Kendall. 2017. Concrete Dropout. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 3581–3590.

[13] Kostadin Georgiev and Preslav Nakov. 2013. A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*. Atlanta, GA, 1148–1156.

[14] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. 2015. MADE: Masked Autoencoder for Distribution Estimation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. Lille, France, 881–889.

[15] Carlos A. Gomez-Uribe and Neil Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manag. Inf. Syst.* 6, 4 (2016), 13:1–13:19.

[16] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*. Perth, Australia, 173–182.

[18] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53.

[19]]higgins2016beta Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. [n. d.]. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. Toulon, France.

[20] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA.

[21] Jun Hu and Ping Li. 2017. Decoupled Collaborative Ranking. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*. Perth, Australia, 1321–1329.

[22] Jun Hu and Ping Li. 2018. Collaborative Filtering via Additive Ordinal Regression. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*. Marina Del Rey, CA, 243–251.

[23] Jun Hu and Ping Li. 2018. Collaborative Multi-objective Ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*. Torino, Italy, 1363–1372.

[24] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*. Pisa, Italy, 263–272.

[25] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[26] Hwichan Kim and Mamoru Komachi. 2021. TMU NMT System with Japanese BART for the Patent task of WAT 2021. In *Proceedings of the 8th Workshop on Asian Translation (WAT@ACL/IJCNLP)*. Online, 133–137.

[27] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA.

[28] Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improving Variational Autoencoders with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems (NIPS)*. Barcelona, Spain, 4736–4744.

[29] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. Banff, Canada.

[30] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[31] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[32] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia.

[33] Ping Li, Christopher J. C. Burges, and Qiang Wu. 2007. McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada, 897–904.

[34] 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 on World Wide Web (WWW)*. Lyon, France, 689–698.

[35] Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*. New York City, NY, 1727–1736.

[36] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep learning recommendation model

for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091* (2019).

[37] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*. Vancouver, Canada, 497–506.

[38] Mathis Petrovich, Michael J. Black, and Gül Varol. 2021. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, Canada, 10965–10975.

[39] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*. Beijing, China, 1278–1286.

[40] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. 2018. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847* (2018).

[41] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada, 1257–1264.

[42] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E. Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML)*. Corvallis, OR, 791–798.

[43] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion (WWW Companion)*. Florence, Italy, 111–112.

[44] Brent Smith and Greg Linden. 2017. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Comput.* 21, 3 (2017), 12–18.

[45] Harald Steck. 2019. Markov Random Fields for Collaborative Filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 5474–5485.

[46] Kevin Swersky, Daniel Tarlow, Ryan P. Adams, Richard S. Zemel, and Brendan J. Frey. 2012. Probabilistic n-Choose-k Models for Classification and Ranking. In *Advances in Neural Information Processing Systems (NIPS)*. Lake Tahoe, NV, 3059–3067.

[47] Ilya O. Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. 2018. Wasserstein Auto-Encoders. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Vancouver, Canada.

[48] Jakub M. Tomczak and Max Welling. 2018. VAE with a VampPrior. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. Playa Blanca, Lanzarote, Canary Islands, Spain, 1214–1223.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, CA, 5998–6008.

[50] 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 (WSDM)*. San Francisco, CA, 153–162.

[51] Jianwen Xie, Zilong Zheng, and Ping Li. 2021. Learning Energy-Based Model with Variational Auto-Encoder as Amortized Sampler. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. Virtual Event, 10441–10451.

[52] Zhiqiang Xu, Dong Li, Weijie Zhao, Xing Shen, Tianbo Huang, Xiaoyun Li, and Ping Li. 2021. Agile and Accurate CTR Prediction Model Training for Massive-Scale Online Advertising Systems. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. Virtual Event, China, 2404–2409.

[53] Mingzhang Yin and Mingyuan Zhou. 2018. Semi-Implicit Variational Inference. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Stockholmsmässan, Stockholm, Sweden, 5646–5655.

[54] 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)*. Macao, China, 4206–4212.

[55] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38.

[56] Xiaofeng Zhang, Jingbin Zhong, and Kai Liu. 2021. Wasserstein autoencoders for collaborative filtering. *Neural Comput. Appl.* 33, 7 (2021), 2793–2802.

[57] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262* (2017).

[58] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2018. The Information Autoencoding Family: A Lagrangian Perspective on Latent Variable Generative Models. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*. Monterey, CA, 1031–1041.

[59] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. 2019. AIBox: CTR Prediction Model Training on a Single Node. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. Beijing, China, 319–328.