VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders

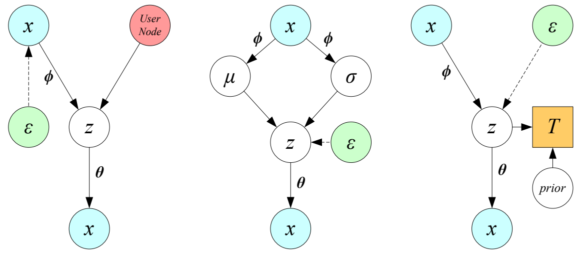
Xianwen Yu[1†], **Xiaoning Zhang**[2*†], **Yang Cao**[2] and **Min Xia**[1]
[1]Department of Software Engineering and Data Technology, Peking University
[2]SenseTime Group Limited
{yuxianwen, mxia}@pku.edu.cn, {zhangxiaoning, caoyang}@sensetime.com

Abstract

Recently, Variational Autoencoders (VAEs) have been successfully applied to collaborative filtering for implicit feedback. However, the performance of the resulting model depends a lot on the expressiveness of the inference model and the latent representation is often too constrained to be expressive enough to capture the true posterior distribution. In this paper, a novel framework named VAEGAN is proposed to address the above issue. In VAEGAN, we first introduce Adversarial Variational Bayes (AVB) to train Variational Autoencoders with arbitrarily expressive inference model. By utilizing Generative Adversarial Networks (GANs) for implicit variational inference, the inference model provides better approximation to the posterior and maximum-likelihood assignment. Then the performance of our model is further improved by introducing an auxiliary discriminative network using adversarial training to achieve high accuracy in recommendation. Furthermore, contractive loss is added to the classical reconstruction cost function as a penalty term to yield robust features and improve the generalization performance. Finally, we show that the performance of our proposed VAEGAN significantly outperforms state-of-the-art baselines on several real-world datasets.

1 Introduction

Collaborative Filtering (CF) technology is one of the earliest and most successful technologies in recommendation systems. In CF, autoencoder is a deep neural network which achieves significant performance and receives much attention recently. As shown in Figure 1(a), Collaborative Denoising Autoencoders (CDAE) [Wu *et al.*, 2016] augments the standard Denoising Autoencoders (DAEs) [Vincent *et al.*, 2008] by adding user latent vector. However, it is prone to overfitting with the increase of both users and items; it also requires additional optimization to obtain user latent vector when dealing with unseen users. Mult-VAE [Liang *et al.*, 2018] successfully applies Variational Autoencoders [Kingma and Welling, 2013] to CF problems. While this model is very flexible in its dependence on the input, the latent variables are often limited to exponential family distributions or other distributions with tractable densities [Rezende and Mohamed, 2015] as shown in Figure 1(b). In fact, using more expressive inference model is essential to make use of the latent space at all [Chen *et al.*, 2016] to get a tighter lower bound which can lead to substantially better results in performing maximum-likelihood training [Kingma *et al.*, 2016]. To develop an expressive inference model, Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014] provide an effective solution.



(a) CDAE (b) VAE (c) AVB

Figure 1: Graphical structures comparison among CDAE, VAE and AVB, where the dashed arrows denote sampling from some noise distributions and T denotes the discriminator.

In GANs framework, two models are simultaneously trained: generative model G, which tends to capture data distribution, and discriminative model D, which tries to estimate the probabilities of sampling from the true distribution. Though GANs have achieved great success in image generation [Brock *et al.*, 2018] and natural language generation [Yu *et al.*, 2016], there is still a lot of room for its application in recommendation systems. The newly proposed IRGAN [Wang *et al.*, 2017] and GraphGAN [Wang *et al.*, 2018] have proved that GANs can also be promising and effective in CF. However, none of the previously proposed GAN-based CF models is combined with autoencoders, which are effective and useful especially when modelling large, sparse, high-dimensional data. To the best of our knowledge, we are the first one to apply GANs to autoencoder structure in CF.

[*] Contact Author
[†] indicates equal contribution.

In this paper, we propose a novel CF framework based on Adversarial Variational Autoencoders, named VAEGAN. The overall structure of VAEGAN is shown in Figure 2. We first introduce Adversarial Variational Bayes (AVB) [Mescheder *et al.*, 2017], which utilizes a flexible black-box inference model. As shown in Figure 1(c), AVB unifies VAEs and GANs through adversarial training. It obtains arbitrarily flexible inference model parameterized by neural networks, which leads to closer approximation to the true posterior distribution for the inference model and approximate maximum-likelihood assignment for the generative model. Then an auxiliary discriminative network is proposed to further reduce reconstruction loss between the generated vector and the ground-truth by adversarial training. Furthermore, we add contractive loss as a penalty term. This term results in a localized space contraction that yields robust features and improves generalization performance. We conduct experiments on several public real-world datasets to evaluate the quality of our proposed framework and investigate the effect of each proposed component. Experimental results show that the performance of VAEGAN significantly outperforms state-of-the-art top-N recommendation methods on some common evaluation metrics.

Our main contributions are summarized as follows:

1. We utilize a flexible black-box inference model as well as adversarial training to train VAEs for implicit variational inference, which unifies VAEs and GANs.
2. We introduce an auxiliary discriminative network to conduct adversarial training to further reduce the reconstruction loss of the observed user-item interactions.
3. Contractive loss is added to the classical reconstruction cost function as a penalty term to yield robust features and improve the generalization performance.

2 Related Work

In this section, we briefly introduce the related works in two aspects. Existing GAN-based and AE-based recommendation methods will be reviewed.

2.1 GAN-based Methods

IRGAN [Wang *et al.*, 2017] proposes a minimax game to iteratively optimize generative model G and discriminative model D. G generates relevant items by sampling from the candidate pool for the given user and D discriminates the ground-truth items from those generated by G. However, owing to the fact that the discrete item index generated by G is probably the same as the ground-truth, D tends to be involved in confusion and gets degraded due to the large portion of contradicting labels for the same item index. To address the problem, CFGAN [Chae *et al.*, 2018] proposes a novel framework. Instead of sampling a single discrete item index, G tries to generate real-valued vectors to prevent D's confusion, which makes D guide G consistently to improve.

2.2 AE-based Methods

DAEs extend the classical autoencoder by training to reconstruct input $\boldsymbol{x}$ from its partially corrupted version $\tilde{\boldsymbol{x}}$. CDAE [Wu *et al.*, 2016] learns latent representation combined with user latent factor from the corrupted user-item preferences and reconstructs the full input. It is prone to overfitting and relatively complicated to scale to unseen users. Mult-VAE [Liang *et al.*, 2018] extends variational autoencoders to CF with multinomial likelihood. The inference model is constrained to produce latent variables that approximately follow some common distributions, and the generative model tends to reconstruct the input by sampling from the distribution. Generally, the inference model may be not expressive enough to capture the true posterior distribution.

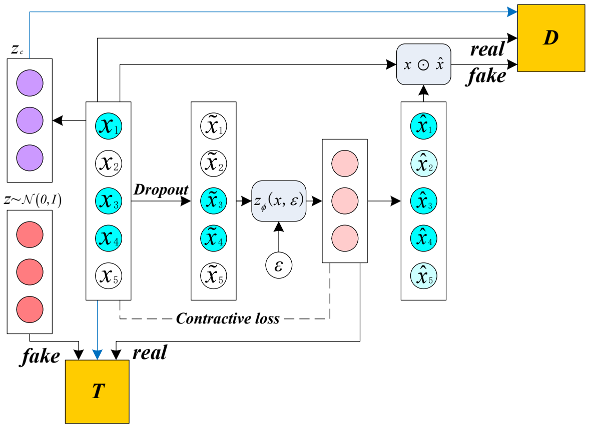


Figure 2: Overall architecture of our method. The blue arrows denote conditional information.

3 Our Method

Given $u \in \{1, ..., M\}$ to index users and $i \in \{1, ..., N\}$ to index items, we define the user-item interaction matrix as $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ from users' implicit feedback. $\boldsymbol{x}_u = [x_{u1}, x_{u2}, ..., x_{uN}] \in \boldsymbol{X}$ denotes the u-th bag-of-words vector, where $x_{ui} = 1$ if the interaction between user u and item i is observed otherwise $x_{ui} = 0$.

3.1 Adversarial Variational Bayes

When performing maximum-likelihood training, it is usually intractable to directly optimize the marginal log-likelihood $\mathbb{E}_{p_{\mathcal{D}}(x)}\log p_\theta(x)$. After using Jensen's inequality, we have:

$$\log p_\theta(x) \geq \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x), p(z)). \tag{1}$$

Variational Bayes rephrases this intractable problem into:

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}[\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x), p(z))]. \tag{2}$$

where $p_{\mathcal{D}}(x)$ is the data distribution and θ, ϕ denote the parameters of generative and inference model respectively.

This is commonly called the variational lower bound or evidence lower bound (*ELBO*). VAEs have an explicit representation of $q_\phi(z|x)$ such as a Gaussian distribution with diagonal covariance matrix whose mean and variance vectors are parameterized by neural networks, which can be optimized using the reparameterization trick [Kingma and Welling, 2013; Rezende *et al.*, 2014] and stochastic gradient descent. With an explicit representation of $q_\phi(z|x)$, it is straightforward to calculate the KL-divergence term $\mathrm{KL}(q_\phi(z|x), p(z))$.

However, explicit variational inference is restrictive w.r.t. its dependence on $\boldsymbol{z}$. A flexible implicit distribution may provide better approximation to the posterior and a tighter lower bound. The resulting latent variables model, which utilizes an implicit likelihood, may fit the data better [Huszár, 2017].

We could try to model the likelihood and the approximate posterior implicitly. Different from Mult-VAE [Liang *et al.*, 2018], we can utilize a flexible black-box inference model $q_\phi(z|x)$ instead. Then adversarial training which unifies VAEs and GANs is introduced to obtain a closer approximation to the real posterior and an approximate maximum-likelihood parameters assignment. While it is intractable to directly obtain the KL-divergence term $\mathrm{KL}(q_\phi(z|x), p(z))$ with an implicit representation of $q_\phi(z|x)$. We can rewrite the optimization problem in (2) as:

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim q_\phi(z|x)}(\log p_\theta(x|z) + \log p(z) - \log q_\phi(z|x)). \tag{3}$$

A discriminative network $T(x,z)$ is introduced to implicitly represent the term $\log p(z) - \log q_\phi(z|x)$ with its optimal value, thus replacing the intractable KL-divergence term $\mathrm{KL}(q_\phi(z|x), p(z))$.

Specifically, given the latent variables representation $q_\phi(z|x)$ and a prior Gaussian representation $p(z)$, the discriminative network $T(x,z)$ is trained to distinguish pairs (x,z) sampled independently using the prior distribution $p_{\mathcal{D}}(x)p(z)$ from those sampled using the posterior distribution $p_{\mathcal{D}}(x)q_\phi(z|x)$. Formally, the objective for the discriminator $T(x,z)$ is as follows:

$$\max_{\Psi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim q_\phi(z|x)}\log\sigma(T(x,z)) + \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim p(z)}\log(1-\sigma(T(x,z))). \tag{4}$$

where $\sigma(t)$ denotes the sigmoid function and Ψ denotes the parameters of the discriminator $T(x,z)$.

The objective in (4) attains the maximum at $\sigma(T^*(x,z)) = \frac{q_\phi(z|x)}{q_\phi(z|x)+p(z)}$, where $T^*(x,z)$ denotes the optimal discriminator. Equivalently, we have

$$T^*(x,z) = \log q_\phi(z|x) - \log p(z). \tag{5}$$

Inserting (5) into (3), the problem can be written as:

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim q_\phi(z|x)}(\log p_\theta(x|z) - T^*(x,z)). \tag{6}$$

Using the reparameterization trick, we reparametrize sampling from q_ϕ in terms of non-linear function z_ϕ and noise variables ε which is generally assumed to be *Gaussian noise*:

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}(\log p_\theta(x|z_\phi(x,\varepsilon)) - T^*(x,z_\phi(x,\varepsilon))). \tag{7}$$

We define the function: $z_\phi(x,\varepsilon) = f(f(\tilde{x}+\varepsilon_1)+\varepsilon_2)$, where f denotes the non-linear function, ε_1 and ε_2 are sampled from gaussian distribution. For simplicity, we use z_ϕ to denote $z_\phi(x,\varepsilon)$ in this paper. The optimization objective (7) has been proved to be optimized directly w.r.t. θ and ϕ using stochastic gradient descent, while it needs to keep $T^*(x,z)$ optimal in contrast. Therefore, we regard the optimization problems (4) and (7) as a two-player minimax game. Additionally, if (θ^*, ϕ^*, T^*) defines a *Nash-equilibrium* for the two-player game defined by (4) and (7), the variational lower bound (*ELBO*) in (2) attains maximum.

3.2 Annealing

The first part of the optimization objective (7) tends to obtain the maximum-likelihood of the reconstructed input $\hat{\boldsymbol{x}}$ from the generative model. We obtain the reconstruction error using the logistic log-likelihood. The second part of (7) can be viewed as a regularization term. In recommendation systems, the CF models are supposed to recommend personalized items which the user might like and hasn't interacted, not to maximize likelihood and generate the accurate reconstructed input. By reducing the constraint of prior distribution on latent variables, we may get better recommendation performance. Meanwhile, simply using the origin regularization term, the model may be over regularized.

Additionally, the optimization of (7) requires to keep the discriminator $T(x,z)$ close to optimality. Instead of performing several SGD-updates for the adversary and one SGD-update for the generative model, it is reasonable to set a parameter α ($0 < \alpha < 1$) to control the strength of the regularization term. The generative model tends to update at a slower pace than the discriminator.

$$\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}(\log p_\theta(x|z_\phi) - \alpha\cdot T^*(x,z_\phi)). \tag{8}$$

When training VAEs, KL annealing [Bowman *et al.*, 2015] and Free Bits [Kingma *et al.*, 2016] are the commonly used methods to ensure the effect of KL-divergence term and prevent KL-vanishing [Bowman *et al.*, 2015] or posterior collapse [Oord *et al.*, 2016]. Similarly, inspired by KL annealing, we adopt a simple heuristic for setting α: we start training with $\alpha = 0$ and gradually increase α, which consistently produces significant recommendation results.

3.3 GAN-based Reconstruction Loss

We introduce an auxiliary discriminative network D to conduct adversarial training to further reduce the reconstruction loss of the observed user-item interactions. The proposed autoencoder in Section 3.1 is regarded as G to generate plausible user-item preference vector that most closely resembles the ground-truth and D, in contrary, tries to distinguish ground-truth from generated vector as accurately as possible. The two-player game makes D guide G consistently to improve so as to generate vectors closer to the ground-truth.

However, it's generally difficult to tell whether a recommendation sequence is real or fake without a specific user. In order to help D to distinguish real input training examples from fake reconstructed input samples, inspired by Conditional GAN [Mirza and Osindero, 2014], a new latent vector $\boldsymbol{z}_c$ is generated to characterize user. The model parameters of D are learned while taking user's personalization into account with the conditional user-personalized vector.

Meanwhile, if we simply regard the reconstructed input $\hat{\boldsymbol{x}}$ as the fake sample, the gradient from D will guide G to reduce the reconstruction loss of both the observed interacted and non-interacted items. In other words, the click probabilities of the interacted items tend to be predicted as 1 and the probabilities of the non-interacted items are assumed to be 0. It seems to make no sense in recommendation tasks as we are more focused on making recommendations on potential preference items which the user has not interacted with rather than obtaining the exact reconstructed results. Therefore, we reset the reconstructed predicting probabilities of the non-interacted items to 0 before we send it into D. In this way, D distinguishes the reconstructed input from the origin input according to the interacted items and G does not get the gradient of the non-interacted items from D. So G has expected predicting probabilities on the non-interacted items, which is reasonable and desirable in recommendation tasks.

Formally, for user u, we take $\{\hat{\boldsymbol{x}} \odot \boldsymbol{x}_u, \boldsymbol{z}_{cu}\}$ as the fake sample and $\{\boldsymbol{x}_u, \boldsymbol{z}_{cu}\}$ as the real sample, where $\odot$ denotes element-wise multiplication and $\{\}$ denotes the concatenation of the vectors inside and $\boldsymbol{z}_{cu}$ is the conditional characterized vector for user u. $\hat{\boldsymbol{x}} \odot \boldsymbol{x}_u$ obtains a processed reconstruction vector where the probabilities of the non-interacted items are reset to 0. The objective for D is as follows:

$$\begin{aligned}&\max_{\tau,\xi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)} \log\sigma(D(x|z_c)) \\ &\quad + \mathbb{E}_{\hat{x}\sim p_\theta(x|z)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)} \log(1-\sigma(D(\hat{x}\odot x|z_c))).\end{aligned} \tag{9}$$

where τ denotes the parameters of the discriminator D and ξ denotes the parameters of the additional part that generating the conditional characterized vectors. The objective for G is:

$$\min_{\theta,\phi} \mathbb{E}_{\hat{x}\sim p_\theta(x|z)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)} \log(1-\sigma(D(\hat{x}\odot x|z_c))). \tag{10}$$

Then we introduce it as a regularization term of the reconstruction loss:

$$\begin{aligned}&\max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}(\log p_\theta(x|z_\phi) - \alpha\cdot T^*(x,z_\phi)) \\ &\quad + \mathbb{E}_{\hat{x}\sim p_\theta(x|z)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)}\beta\cdot\log\sigma(D(\hat{x}\odot x|z_c)).\end{aligned} \tag{11}$$

where β is a tunable parameter to control the strength of the regularization term.

3.4 Contractive Loss

Though the inference model is theoretically flexible and expressive, the latent space may be over expanded leading to a delicate recommendation model. It has been demonstrated that many state-of-the-art classifiers are actually very fragile and vulnerable to adversarial examples [Szegedy *et al.*, 2013], implying that the model is prone to suffering from the perturbations of the input. Inspired by Contractive Auto-Encoders [Rifai *et al.*, 2011], we propose a simple and effective way to obtain robust latent representations.

We add a penalty term as the contractive loss to the reconstruction objective function. This penalty term corresponds to the *Frobenius norm* of the Jacobian matrix of the latent variables from the inference model with respect to the input, resulting in a localized space contraction thus yields robust representations. As a result, the latent representations tend to be robust to small changes of the input. In recommendation systems, it is common to obtain exactly similar input from different users as the input data is often large and sparse. By introducing contractive loss, we can get exactly similar latent representations with respect to similar input and the generalization performance of the model can be improved.

Formally, in our model, the input $\boldsymbol{X} \in \mathbb{R}^{M\times N}$ is mapped by the inference model $q_\phi(z|x)$ to the latent representations $\boldsymbol{z}$. We utilize a sensitivity penalization term to penalize its sensitivity to the input, which is the sum of squares of all partial derivatives of the latent representations w.r.t. input:

$$\|J_{q_\phi}(\boldsymbol{X})\|_F^2 = \sum_{u=1,i=1}^{M,N}(\frac{\partial z_u}{\partial x_{ui}})^2. \tag{12}$$

where z_u denotes the latent representation of user u. The mapping to the latent space is encouraged to be contractive in the neighborhood of the training data by penalizing $\|J_{q_\phi}(\boldsymbol{X})\|_F^2$, implying an invariance or robustness of the latent representations for small changes of the input.

Overall, our final objectives of VAEGAN are as follows:

$$\begin{aligned}J^G &: \max_{\theta,\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}(\log p_\theta(x|z_\phi) - \alpha\cdot T^*(x,z_\phi)) \\ &\qquad + \mathbb{E}_{\hat{x}\sim p_\theta(x|z)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)}\beta\cdot\log\sigma(D(\hat{x}\odot x|z_c)) \\ &\qquad + \lambda\cdot\|J_{q_\phi}(\boldsymbol{X})\|_F^2, \\ J^T &: \max_{\Psi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim q_\phi(z|x)}\log\sigma(T(x,z)) \\ &\qquad + \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z\sim p(z)}\log(1-\sigma(T(x,z))), \\ J^D &: \max_{\tau,\xi} \mathbb{E}_{x\sim p_{\mathcal{D}}(x)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)} \log\sigma(D(x|z_c)) \\ &\qquad + \mathbb{E}_{\hat{x}\sim p_\theta(x|z)}\mathbb{E}_{z_c\sim q_\xi(z_c|x)} \log(1-\sigma(D(\hat{x}\odot x|z_c))).\end{aligned} \tag{13}$$

where λ denotes a tunable parameter to control the strength of penalizing the contractive loss.

4 Experiments and Analysis

In this section, we perform thorough experiments to evaluate our proposed VAEGAN on four real-world datasets. We aim to answer the following research questions:

Q1. How does our proposed method perform compared with the state-of-the-art top-N recommendation methods?

Q2. How much does each of our proposed components contribute to the model?

Q3. How much is our proposed method influenced by the key hyperparameters?

4.1 Experiments

Datasets

In order to fully demonstrate the effectiveness of our proposed VAEGAN, we study the performance of different models under both strong and weak generalization [Marlin, 2004]. Four common real-word datasets: MovieLens-20M, Nextflix Prize[1], MovieLens-1M and MovieLens-100K are used to evaluate our method. In strong generalization experiments, we use two medium to large scale datasets: MovieLens-20M and Netflix Prize as the Mult-VAE [Liang *et al.*, 2018] did. All

[1] https://www.netflixprize.com/

Metrics	R@5	R@20	R@50	G@10	G@20	G@100	M@10	M@20	M@100
Dataset	MovieLens-20M								
Mult-VAE	0.308	0.395	0.537	0.318	0.334	0.426	0.511	0.516	0.518
Mult-DAE	0.311	0.387	0.524	0.311	0.331	0.419	0.515	0.520	0.522
aWAE	0.326	0.391	0.532	0.338	0.338	0.424	-	-	-
Ours-AVB	0.335	0.402	0.539	0.337	0.349	0.434	0.542	0.547	0.549
Ours-AVB+D	**0.339**	0.405	0.539	0.341	0.352	0.436	0.546	0.551	0.553
Ours-AVB+D+C	**0.339**	**0.407**	**0.541**	**0.343**	**0.354**	**0.438**	**0.549**	**0.554**	**0.556**
Dataset	Netflix								
Mult-VAE	0.324	0.355	0.444	0.331	0.320	0.386	0.510	0.515	0.517
Mult-DAE	0.321	0.344	0.438	0.317	0.314	0.380	0.504	0.509	0.511
aWAE	0.301	0.354	0.441	0.256	0.331	0.381	-	-	-
Ours-AVB	0.353	0.360	0.445	0.346	0.337	0.393	0.544	0.548	0.551
Ours-AVB+D	0.354	0.362	0.445	0.347	0.339	0.394	0.547	0.552	0.554
Ours-AVB+D+C	**0.355**	**0.363**	**0.447**	**0.349**	**0.340**	**0.396**	**0.549**	**0.554**	**0.556**

Table 1: Strong generalization experiments on MovieLens-20M and Netflix. Best results are shown in **bold**.

users are divided into training/validation/test sets. Specifically, we take 10K and 40K users as the the held-out validation/test users for MovieLens-20M and Netflix Prize respectively. The entire click histories of the training users are used to train models. For evaluation, we take 80% of the click histories for each held-out user as the input and compute metrics on the remaining 20% of the click histories. The explicit data is binarized by keeping those with ratings no less than 4 stars and only users who have rated at least five movies are kept. In weak generalization experiments, we use two small datasets: MovieLens-1M and MovieLens-100K as the CF-GAN [Chae *et al.*, 2018] did. We split all the user-item interactions of each user into two subsets: 80% for training and 20% for testing. All the explicit data is treated as implicit feedback and we only keep users who have rated at least five movies.

Evaluation Metrics

To evaluate our model, we adopt three common metrics for top-N recommendation: recall(R@N), normalized discounted cumulative gain(G@N), and mean reciprocal rank(M@N). While Recall@N equally weights all the top-N items, NDCG@N and MRR@N assign higher scores to higher ranks.

Baselines

We compare our proposed VAEGAN with the following state-of-the-art collaborative filtering methods:

CDAE. It extends the denoising autoencoders (DAEs) by adding user-specific latent vector [Wu *et al.*, 2016].

IRGAN. It is the pioneer GAN-based method that successfully applies GANs to CF [Wang *et al.*, 2017].

CFGAN. It suggests a new direction of vector-wise adversarial training [Chae *et al.*, 2018].

Mult-VAE and Mult-DAE. They improve the performance of variational autoencoders (VAEs) and denoising autoencoders (DAEs) by using a multinomial likelihood for the data distribution [Liang *et al.*, 2018].

aWAE. It extends the Wasserstein Autoencoders [Tolstikhin *et al.*, 2017] for collaborative filtering to address the problem that the distributions of the encoder latent variables overlap a lot [Zhong and Zhang, 2018].

Some of the above methods, such as CDAE, are not suitable for training under strong generalization since they lack the necessary latent representations for unseen held-out users. Though we could reluctantly solve this problem using additional optimization, it is not a rigorous and precise way to verify models. For fair comparisons, we train them under weak generalization. Meanwhile, training models under strong generalization can better illustrate the effectiveness and robustness, as it's relatively more difficult to train models under strong generalization than weak generalization. So, we train the other models under strong generalization.

Implementation Details

We keep the model with the best validation NDCG@N and the structure and hyperparameters are dertermined according to test metrics with it. Concretely, following Mult-VAE, we adopt symmetrical autoencoder structure and set the dimension of the bottleneck layer to 200. The overall model structure is $[I \rightarrow 600 \rightarrow 200 \rightarrow 600 \rightarrow I]$, where I denotes the total number of items. We apply dropout [Srivastava *et al.*, 2014] at the input layer with probability 0.5 to avoid overfitting. Weight decay isn't adopted in our model as we already have regularization terms. We train our model using Adam with batch size 500 and 256 under strong and weak generalization, respectively. We train for 300 epochs for MovieLens-20M, 200 epochs for Netflix Prize and 1000 epochs on the other two datasets. Other key hyperparameters will be further discussed in Section 4.4.

4.2 Q1. Comparisons with The State-of-the-arts

In this section, we compare our proposed VAEGAN with several popular top-N recommendation methods under strong and weak generalization as discussed in Section 4.1.

Table 1 shows the results of various models under strong generalization on MovieLens-20M and Netflix, respectively. We can see that our model achieves the best performance on both datasets. Our methods outperform significantly than Mult-VAE, which is the most similar method to ours, in terms of all the metrics. It demonstrates that AVB improves the restricted performance of VAE by relieving the problems that the latent representations are limited and the inference model is not expressive enough. Additionally, our model achieves

Metrics	R@5	R@20	G@5	G@20	M@5	M@20	R@5	R@20	G@5	G@20	M@5	M@20
Dataset	MovieLens-100K						MovieLens-1M					
CDAE	0.144	0.353	0.465	0.425	0.664	0.674	0.108	0.272	0.439	0.401	0.629	0.644
IRGAN	0.107	0.275	0.342	0.368	0.536	0.523	0.072	0.166	0.264	0.246	0.301	0.338
CFGAN	0.152	0.360	**0.476**	0.433	0.681	0.693	0.108	0.272	0.455	0.406	0.647	0.660
Ours-AVB	0.151	0.362	0.459	0.433	0.686	0.698	0.113	0.279	0.460	0.414	0.656	0.669
Ours-AVB+D	**0.153**	**0.365**	0.467	**0.437**	**0.688**	**0.700**	**0.115**	**0.281**	**0.465**	**0.416**	**0.663**	**0.676**
Ours-AVB+D+C	0.152	0.364	0.468	0.436	**0.688**	**0.700**	0.114	**0.281**	0.464	**0.416**	0.662	0.674

Table 2: Weak generalization experiments on MovieLens-1M and MovieLens-100K. Best results are shown in **bold**.

superior accuracy by the adversarial training on the generated recommendation results and the contractive regularization term resulting in robust features. Compared to other non-linear AE-based methods, our model imposes stronger modelling assumptions and achieves state-of-the-art performance.

Table 2 shows the results under weak generalization on MovieLens-1M and MovieLens-100K. We can see our model outperforms the state-of-the-art methods. Compared to other GAN-based methods, our model both relies on the adversarial training process of GANs and combines the AE structure which is effective and powerful when user-item interactions are scarce. Overall, our proposed VAEGAN consistently produces good results under both strong and weak generalization compared with GAN-based and AE-based methods.

4.3 Q2. Effectiveness of Proposed Components

AVB. It has been illustrated that AVB obviously performs better than Mult-VAE in Table 1, 2. AVB utilizes a flexible black-box inference model and adversarial training to enhance the expressiveness of the model, thus obtaining a better approximation to the true posterior distribution.

AVB+D. Table 1, 2 show that after adding an auxiliary discriminative network D, the model performs better. We conduct additional experiments to study the influence of the adversarial regularization coefficient in Section 4.4. The discriminative network D implicitly measures the distance between distributions of input data and reconstructed data using black-box neural networks. The two distributions are getting closer and closer through adversarial training.

AVB+D+C. Table 1 shows that the model performs better after introducing the contractive regularization term. We also conduct additional experiments to study the influence of the contractive regularization coefficient in Section 4.4. The penalty term encourages the latent representations to be robust to small changes of the input around the input examples. It would be effective to learn robust latent representations when modelling unseen click histories. However, under weak generalization, the training input examples also appear during evaluation. The contractive regularization term can hardly play its due role under weak generalization as the experimental results show in Table 2.

4.4 Q3. Influence of The Key Hyperparameters

In this section, we mainly investigate the following hyperparameters in our proposed method: the annealing coefficient (α), regularization coefficient of D (β), contractive regularization coefficient (γ). Figure 3 shows the validation NDCG@100 with and without annealing to $\alpha = 0.2$ and 1 (α reaches maximum at around 80 epochs). We can see that with annealing the model converges faster and obtains better performance. In this paper, we set α as 0.2 which consistently produces good results on the test set. Figure 4 shows the performance of our method depending on the value of β and γ. As we can see, when we set β as 0.2, γ as 0.0003, the proposed method reach the best results. The model may be over regularized or the proposed component may be underutilized with other parameters settings.

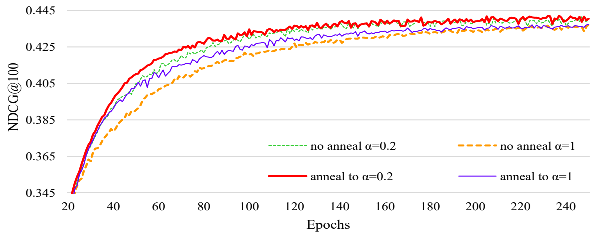
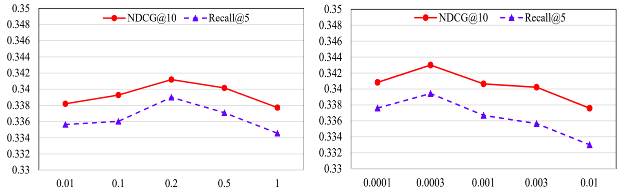


Figure 3: Analysis of annealing parameter α.



(a) β (b) γ

Figure 4: Analysis of parameter β and parameter γ.

5 Conclusions

In this paper, we present a variant of Variational Autoencoders based on adversarial training for collaborative filtering. The inference model is much more flexible and expressive to model almost arbitrary posterior distributions over the latent variables. GAN-based reconstruction loss further improves the performance of our model. Moreover, we obtain robust latent representations by introducing contractive loss.

Acknowledgments

The authors would like to thank the Knowledge Discovery and Behavior Analysis Department of SenseTime Research for the valuable support.

References

[Bowman *et al.*, 2015] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[Brock *et al.*, 2018] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[Chae *et al.*, 2018] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 137–146, 2018.

[Chen *et al.*, 2016] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Huszár, 2017] Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Kingma *et al.*, 2016] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[Liang *et al.*, 2018] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 27th International Conference on World Wide Web*, pages 689–698, April 2018.

[Marlin, 2004] Benjamin Marlin. *Collaborative filtering: A machine learning perspective*. University of Toronto, 2004.

[Mescheder *et al.*, 2017] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2391–2400, 2017.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[Oord *et al.*, 2016] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[Rezende and Mohamed, 2015] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1278–1286, 2014.

[Rifai *et al.*, 2011] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840, 2011.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Tolstikhin *et al.*, 2017] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine Learning*, pages 1096–1103, 2008.

[Wang *et al.*, 2017] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 515–524, 2017.

[Wang *et al.*, 2018] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 2508–2515, 2018.

[Wu *et al.*, 2016] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162, 2016.

[Yu *et al.*, 2016] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu Seqgan. sequence generative adversarial nets with policy gradient. arxiv preprint. *arXiv preprint arXiv:1609.05473*, 2(3):5, 2016.

[Zhong and Zhang, 2018] Jingbin Zhong and Xiaofeng Zhang. Wasserstein autoencoders for collaborative filtering. *arXiv preprint arXiv:1809.05662*, 2018.