Mix Variational Autoencoder for Rating Prediction

Jinxin Liu[1,2], Yingyuan Xiao[1,2], Ke Zhu[1,2], Wenguang Zheng[1,2], Ching-Hsien Hsu[3]
[1] *Engineering Research Center of Learning-Based Intelligent System, Ministry of Education, Tianjin, China*
[2] *Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin, China*
[3] *Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan*
jinxinliu@aliyun.com;{yyxiao;wenguangz}@tjut.edu.cn; robertchh@asia.edu.tw

***Abstract*—In recent years, Variational AutoEncoder (VAE) based methods have made many important achievements in the field of recommendation systems. VAE is a kind of Bayesian model which combines latent variable model with variational inference, but its optimization is often troubled by posterior collapse. By comparing the optimization process of VAE and ordinary autoencoder, we observe that the mismatch between poorly optimized encoder and decoder with too strong characterization capabilities makes it difficult to learn the mapping from the data manifold to the parameterized graph. Since the learning of a posteriori network corresponds to the encoder, we think that the problem of a posteriori collapse can be alleviated by balancing the encoder and decoder better. Therefore, we proposed MixVAE, which combines conventional VAE with deterministic autoencoder, and has the advantages of both VAE and deterministic autoencoder. Experiments on three real-world recommendation data sets show that our method alleviates the posterior crash problem in VAE and improves the recommendation performance.**

***Index Terms*—Recommendation System, Autoencoder, Rating Prediction, Deep Learning**

I. Introduction

In the past few years, generative model methods based on deep learning have gained more and more attention due to some amazing achievements. With the help of massive data, well-designed networks, and intelligent training technologies, deep generation models have been able to generate highly realistic content fragments, such as images and texts. The variational autoencoder (VAE) [1] is a generation model combining deep learning with Bayesian inference. VAE belongs to the autoencoder model in structure, but different from the conventional autoencoder, it infers posterior distribution for each instance and optimizes ELBO by sharing inference network. The coding distribution is regularized in the training process to ensure that its potential space meets certain characteristics, and at the same time, it allows us to perform biased sampling from space to generate some new data.

Considering the powerful feature extraction capabilities of VAE, people are very interested in using VAE to model a recommendation system: Given the interaction records of users and items, can we use VAE to generate new interaction records that have not yet occurred as recommended content? In fact, a method using VAE with multinomial likelihood (Mult-VAE) has been proven to be very effective for collaborative filtering [2], and it surpassed the most advanced methods at that time. Then, another method called MacridVAE [3] decouples the latent space, and after layering the data, each group corresponds to a separate VAE, which further improves the recommended effect of VAE.

Although these conventional VAE-based methods have been proven to be effective recommendation schemes, there are still some important limitations. For example, posterior collapse, which is common in Bayesian models, also exists in VAE. Some previous work has observed that the training optimization of VAE is often troubled by the collapse of the posterior probability, that is, the posterior probability and prior probability are almost the same, and the model cannot learn new information to update itself after inputting new samples. In fact, there are many unknowns about the hidden spatial structure of VAE, and the construction of the hidden spatial structure is directly related to the goal and task. In the past few years, people have aroused great interest in exploring the hidden spatial structure of VAE [4] [5]. Another problem is that the potential features in the recommendation system have their particularities. Some exploratory experience in the field of image and text generation may not be suitable for recommending tasks. So, what kind of VAE structure is suitable for the recommendation system? Solving these problems is a promising direction for improving the recommendation system because it will benefit from the modeling ability of deep learning and the inferencing ability of the Bayesian model.

In order to achieve this goal, we will improve the model structure of VAE, so as to alleviate the problem of posterior collapse and make it more in line with the recommendation task. A widely mentioned explanation for posterior collapse is that the strong decoder in VAE makes the disappeared posterior fall into the local optimum of ELBO. To address this issue, existing solutions include weakening the decoder [6], modifying the regular term [7], replacing the posterior family [8] and improving the optimization strategy [9]. In this paper, we look at this problem from another angle. By comparing the optimization process between VAE and ordinary autoencoder, we can observe the mismatch between poorly optimized encoder and decoder with strong performance. From the point of view of differential geometry, this problem shows that the mapping from the data manifold to parametric graph is poor, which makes it difficult to learn the conversion graph

Yingyuan Xiao and Ke Zhu are corresponding authors.

between them [10]. As the posterior network is a part of the VAE transformation graph, corresponding to the encoder part (the whole generating model can be regarded as a graph), we think that the posterior collapse can be alleviated by better parameterization.

For this reason, we propose the MixVAE model, which combines the VAE model with a deterministic autoencoder structure. The new encoder includes a deterministic encoder and a VAE encoder. In addition, in order to better coordinate the encoders, we contributed some encoder weight between the two branches. Experiments on three real-world data sets show that our method alleviates the posterior collapse problem in VAE. Our contributions are as follows:

- We observed the mismatch between encoder and decoder, and then connected it with the posterior collision problem.
- We put forward MixVAE, which helps the encoder and decoder of VAE achieve balance in the training process by adding a deterministic encoder. At the same time, the combined encoder greatly expands the original hidden spatial structure of the deterministic encoder. The new model has the advantages of both encoders.
- Experiments on MovieLens, Douban, and Flixster show that the proposed method improves the performance of the VAE model in the recommendation system.

II. Conventional VAE for Collaborative Filtering

Fig. 1 presents a conventional structure of VAE. On the left is the input x_u, which is encoded to obtain intermediate variable and then decoded to obtain the predicted output on the right. Following the experience in [11] , we set the input of the VAE model as a bag-of-items vector instead of a bag-of-users vector, and each input is the historical records of one user. We define the variable $u \in 1, \ldots, U$ to index the users and $i \in 1, \ldots I$ to index the items. The original data set is converted into a user-item matrix, which is defined as $X \in \mathbb{N}^{U \times I}$. Each raw N^I in X is associated with a user x_u, which is a "bag of items" vector (i.e. a bag of words vector over items). For example, in the case of online rating data, x_u represents the rating of user u. Generally, not every item is rated by the user, and items without a rating will be set to zero or the average value of the user rating.

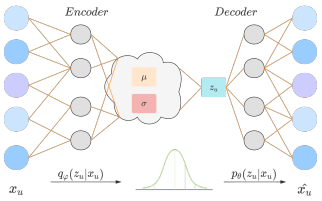

Fig. 1. Architecture of Variational Autoencoder.

VAE for CF [2] is a probability generation model that wants to extract latent representation from the user-item matrix X in high dimension into a low dimensional space. Different from other embedding-based methods, the latent representation in VAE is a probability distribution. Usually, in VAE model we assemble the distribution a normal distribution [12], since it a normal distribution we can use two parameters mean μ and variance σ to describe it. Parameterization of the probability distribution is also beneficial to the subsequent calculation of backpropagation using the technique of reparameterization. Then, the decoding part first samples from the distribution, and then obtains the prediction output through the decoder. Next, we will describe in detail the generation process, inferencing process, and learning objective of VAE.

A. Generation Process

The generation process we consider here is similar to the deep latent Gaussian model [13]. The generation process begins with randomly sampling a K-dimensional potential representation z_u from a standard Gaussian prior with zero mean and identity covariance matrix. The formula is as follows:

$$z_u \sim \mathcal{N}(0, I_k). \tag{1}$$

Next, the z_u will be transformed to a probability distribution vector $\pi(z_u)$ over I items by a non-linear function $f_\theta \in \mathbb{R}^I$ followed by the softmax function (the output of softmax can be regarded as probability distribution whose sum is 1). The f_θ here is a multilayer perceptron with a parameter θ. And $\pi(z_u)$ is a simplex of $\mathbb{S}^{(I-1)}$ over the whole item set [14]. We have:

$$\pi(z_u) \propto exp f_\theta(z_u). \tag{2}$$

Given the total number of clicks $N_u = \sum_i x_{u_i}$ from user u, the observed bag-of-words vector x_u is assumed to be sampled from a multinomial distribution with probability $\pi(z_u)$, as shown in Formula 3. It is worth noting that the common latent factor model is a special case of the current generation model: we can restore the classical matrix factorization [15] by setting $f_\theta(\cdot)$ to linear and using Gaussian likelihood.

$$x_u \sim Mult(N_u, \pi(z_u)). \tag{3}$$

The log-likelihood for user u (conditioned on the latent representation) is:

$$\log p_\theta(x_u|z_u) = \sum_i x_{ui} \log \pi_i(z_u). \tag{4}$$

This multinomial likelihood is often used in language models, e.g., latent Dirichlet allocation, and economics, e.g., multinomial logit choice model. It can also be used for cross-entropy loss in multi-class classification. Unlike Gaussian likelihood, which is widely used in image generation tasks, multinomial likelihood is very suitable for recommendation tasks and shows its ability in practical work [2].

B. Inference Model

We need to estimate the value of θ in $f_\theta(\cdot)$, which is associated with the input data. For this reason, we approximate the intractable posterior distribution $p_\theta(z_u|x_u)$ by using variational inference for each user. Variational inference uses a

simpler distribution $q(z_u)$ to approximate the true intractable posterior. $q(z_u)$ is set as Gaussian distribution with independent components (covariance matrix is a diagonal matrix):

$$q(z_u) = \mathcal{N}\left(\mu_u,\ diag\{\sigma_u^2\}\right). \tag{5}$$

The goal of variational inference is to optimize the free variational parameters $\mu_u,\ \sigma_u^2$, but there is a problem: μ_u and σ_u^2 can have any dimension. Although we can use Monte Carlo methods to estimate, it is unrealistic in the actual recommendation system, because the calculation cost is unacceptable. Therefore, VAE replaces the estimation of individual variational parameters with data-dependent function (usually called inference model) [1]. Now the Formula 5 can be written as:

$$q_\phi(z_u|x_u) = \mathcal{N}\left(\mu_\phi(x_u), diag\{\sigma_\phi^2(x_u)\}\right). \tag{6}$$

Both $\mu_\phi(x_u)$ and $\sigma_\phi(x_u)$ are K dimension vectors, ϕ is a parameter of posterior distribution computed by the inference model (encoder).

C. Training VAEs

Followed the standard process of learning latent variable models with variational inference, we can lower-bound the log marginal likelihood of the data. This means that, for each user u, the target can be converted into maximum the evidence lower bound (ELBO), as shown in:

$$\begin{aligned}\mathcal{L}(x_u;\theta,\phi) \equiv\ & \mathbb{E}_{q_\phi}(z_u|x_u)\left[\log p_\theta(x_u|z_u)\right] \\ & - KL\left(q_\phi(z_u|x_u)\,||p(z_u)\right) \\ & \leq logp(x_u;\theta).\end{aligned} \tag{7}$$

In Formula 7, the first term can be seen as the reconstruction error and the second term is the distance between prior distribution and posterior distribution (KL divergence). With this regular term, the two distributions will be trained to get closer. Another important improvement of VAE is called β-VAE [16], which add a parameter β to ELBO:

$$\begin{aligned}\mathcal{L}_\beta(x_u;\theta,\phi) \equiv\ & \mathbb{E}_{q_\phi}(z_u|x_u)\left[\log p_\theta(x_u|z_u)\right] \\ & -\beta * KL\left(q_\phi(z_u|x_u)\,||p(z_u)\right).\end{aligned} \tag{8}$$

The addition of item β alleviates the problem of a posteriori disappearance to some extent.

D. Rating Prediction

Given the user u, the encoder $(q_\phi(z_u|x_u))$ will encode the user's interaction record x_u into a probability distribution space with parameters μ_ϕ and σ_ϕ^2. Then, the vector z_u is sampled from this space. A new vector $\hat{x_u}$ is a reconstruction as a prediction over the item via the decoder $(p_\theta(x_u|z_u))$. By optimizing Equation 8, it is expected that the probability quality of this model is more in line with the user's preference. That is, the results predicted by the model are more consistent with the actual preferences of users in distribution.

III. DeepAE for Collaborative Filtering

Another important type of autoencoder structure we need to mention here is Deep Autoencoder [17], which is referred to as DeepAE for short. It extends the original three-layer autoencoder [11] to the deep autoencoder. It uses MMSE as a loss function, which is the same as AutoRec. And the use of activation functions with non-zero negative part and unbounded positive part was found to be better. In addition, it also uses dropout layers after the potential layer to avoid overfitting. It also shows that using a large dropout rate after the potential layer can enable it to learn robust representations.The following Fig. 2 shows the structure of DeepAE:

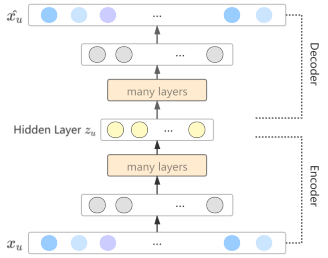

Fig. 2. Architecture of Deep Autoencoder.

A. Training Steps of DAEs

Compared with VAE, the training process of DAE is much more intuitive. Given the historical record x_u of user u as input, the encoder encodes x_u into hidden vector z_u, and then the decoder decodes z_u to obtain $\hat{x_u}$. Calculate the difference between the non-zero part of the original input data x_u and the value of the corresponding position in the prediction $\hat{x_u}$.The optimization goal is to reduce the difference between the two parts. After the model training is finished, an interactive record x_i is input for prediction. Those parts of the original input x_i that are zero have now been filled with corresponding values, and these are the prediction results.

IV. MixVAE

Optimization of VAE is a challenging task. Although the previous β-VAE method has alleviated the problem of posterior disappearance to some extent, there are still many places worth studying in improving the objective optimization of VAE. In order to understand the problem of posterior collapse, we need to have a deeper understanding of VAE training dynamics. Here we compare the loss functions of VAE and DeepAE in the reconstruction process, as shown in the Fig. 3, 4:

We can find that the reconstruction ability of VAE and DeepAE is constantly improving at the beginning of training, and as the number of training continues to increase, VAE has fallen into a bottleneck, but DeepAE is still being optimized slowly. However, there are serious problems in DeepAE itself. When not using additional optimization techniques, it is easy to overfit when the encoder and decoder are too powerful and the depth of DeepAE is too deep. Because the autoencoder is a

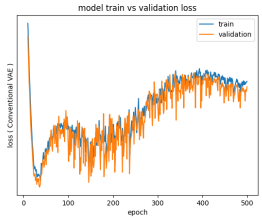

Fig. 3. Training loss of VAE.

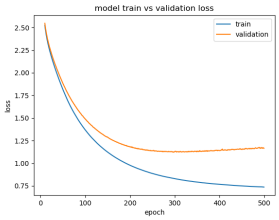

Fig. 4. Training loss of DAE.

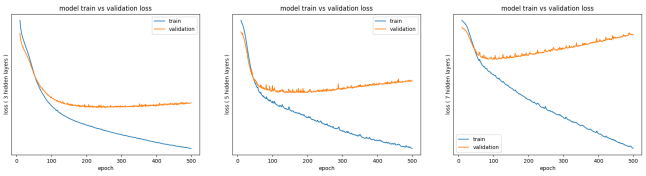

Fig. 5. Training situation of DeepAE with different layers.

kind of compression structure, it will force the model to learn high-dimensional potential features during the compression process. When the fitting ability of the model is far greater than the demand of data, the learning process of the model will become "lazy". Fig. 5 shows the result of increasing hidden layer of DeepAE. The number of hidden layers in three figures in Fig. 5 is 3, 5, 7, respectively.

It can be seen that the loss of the training set is declining, but the loss of the verification set starts to rise at a certain time point, and with the increase of the number of model layers, the turning point of the rise is more forward. That is to say, the deeper the number of layers is, the earlier the model starts over-fitting.

The comparison of the above training processes shows that DeepAE training reconstruction is more stable than VAE to some extent, but VAE has advantages in generalization ability and generation diversity. So, can we combine VAE training with DeepAE? According to this idea, we extend the VAE model described in section II with DeepAE and propose our model MixVAE. MixVAE adds deterministic encoder from DeepAE to the encoder of VAE. The two parts share the same input and combine the output prediction. In this body of work, we focus on alleviating the problem of posterior disappearance in VAE-based recommendation system, but our method can be easily extended to other types of VAE-based tasks.

A. Model Structure

Our proposed model branches the input of the conventional VAE to a deterministic autoencoder, and then the results of the deterministic autoencoder are combined with the branch of VAE at the back of the model. This modification of VAE makes the model generation no longer depend on random sampling in the prior space $p(z)$, which enhances the stability of the model in this way. And it is promising to further mitigation the posterior collapse in VAEs [18]. The overall model structure is shown in the Fig. 6:

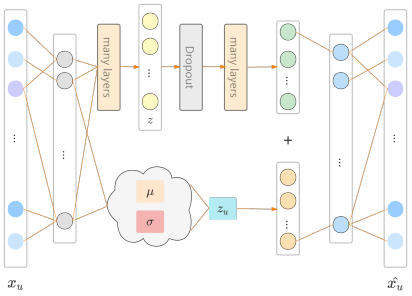

Fig. 6. Architecture of MixVAE.

The input of the model is x_u, which is pre-coded through a full connection layer, and then the model is divided into two branches corresponding to VAE and DeepAE. The VAE part is still connecting two vectors μ and σ with the same dimension which represent parameters of Gaussian distribution. Then, a new vector z_u is obtained by reparameter sampling, and the dimension of z_u is the same as that of the input layer after passing through the decoding layers. In the part of DeepAE, the embedded representation z is obtained after the precoding layer and multiple fully connected layers with L2 regularization, followed by the Dropout layer. Adding a dropout layer can prevent the model from over-fitting. At each iteration, neurons are randomly output to zero according to the specified ratio. After that, the DeepAE branch has been decoded by multiple layers to get the dimension consistent with the input layer. Then the vectors of the two branches are spliced together and pass through a full connection layer as the final output $\hat{x_u}$. Under such a mixed model, the original optimization goal of VAE also changes. The extended ELBO becomes:

$$\mathcal{L} \equiv \mathcal{L}_{rec} + \beta * \mathcal{L}_{KL} + \mathcal{L}_{Deep} \tag{9}$$

Formula 9 adds optimization constraints to the DeepAE part. This modeling method is simple and effective. The addition of DeepAE encoder has alleviated the dilemma that the encoder is significantly weaker than the decoder in the original VAE to a certain extent, and the balance between them has been re-established.

V. EMPIRICAL STUDY

A. Data Sets

In order to evaluate the effect of the model, we used three general data sets, namely MovieLens-1M, Douban data set and Flixster data set. MovieLens data set is one of the most popular movie data sets in the recommendation system, which contains the user's ratings on movies, ranging from 1 to 5. Douban is a Chinese website for exchanging views on books and movies. The Douban 3k data set [19] used here contains about 3,000 users' ratings for 3,000 items, and the users' ratings are also an integer from 1 to 5. The Flixster data set is also provide by Monti [19], the number of users and items is also about 3,000, and the score record is from 0.5 to 5. The following table I shows detailed statistics:

TABLE I
STATISTICS OF DATA SETS.

	MovieLens-1M	Douban	Flixster
users	6,040	3,001	2,999
items	3,952	3,000	2,998
interactions	1,000,209	136,891	26,173
Sparsity	95.81%	98.48%	99.71%

B. Experimental Setting

In terms of experimental setup, we have adopted a similar scheme to our predecessors [2]. We use "Stratified Shuffle Split" function in scikit-learn to divide the data into training set, verification set and test set, with the proportions of 80%, 10% and 10% respectively. In the whole training process, only the training set is involved in the optimization iteration of the model. In the part of verification and evaluation, the deviations between the non-zero parts of the verification set and the test set and the predicted values is compared.

For our model, we use symmetrical structure in VAE substructure, the input dimension I is the number of items, the middle-hidden dimension is 256-512-256, and the result dimension generated by VAE is still I. Then, it is spliced with the output of DAE, and after passing through a perceptron, it outputs the rating prediction of I items. The activation function in the model is RELU, the regularization uses L2 term, the regularization coefficient is 0.0001, the dropout value of DAE part is 0.8, and the β coefficient of VAE part is set to 0.2. The batch size of training is set to 500, and the optimizer uses RMSprop with a coefficient of 0.0001.

C. Metrics

For evaluation, we use the general evaluation indexes Masked Mean Absolute Error (MMAE) and Masked Root Mean Square Error (MRMSE) in the recommendation system. MASK here refers to filtering rules. In the concrete calculation, we compare the deviation between the actual scores of users and the predicted values in the test set, but the parts where the users did not give scores is not included. The calculation formula of both is as follows:

$$MMAE = MASK * \frac{1}{m}\sum_{i=1}^{m}\left|h\left(x^i\right) - y^i\right|, \quad (10)$$

$$MRMSE = MASK * \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(x^i\right) - y^i\right)^2}. \quad (11)$$

D. Baselines

We compare our proposed method which we called MixVAE with the following baselines. Some comparative experiments are based on the algorithm scheme of RecBole recommendation system [20], which is an open source computing platform, and we have made some modifications to meet the rating prediction task.

TABLE II
COMPARISON OF EXPERIMENTAL RESULTS OF VARIOUS METHODS (MMAE AND MRMSE)

Model	MOVIELENS-1M		DOUBAN-3K		FLIXSTER-3K	
	MMAE	MRMSE	MMAE	MRMSE	MMAE	MRMSE
RAND	1.598	1.918	1.865	2.624	2.119	2.965
MacridVAE	1.142	1.159	1.095	1.379	1.232	1.176
Mult-DAE	0.768	0.921	0.731	0.846	0.830	0.933
CDAE	0.770	0.839	0.628	0.780	0.927	0.932
Mult-VAE	0.816	0.864	0.764	0.877	0.823	0.927
I-AutoRec	0.781	0.842	0.633	0.772	0.816	0.910
RecVAE	0.778	0.856	0.853	0.744	0.945	0.908
MixVAE	**0.764**	**0.807**	**0.608**	**0.738**	**0.787**	**0.874**

- Random (RAND). Randomly generate a rating from the original scoring options of the data set.
- Autoencoders meet collaborative filtering (I-AutoRec) [11]. This paper is the first work to use self-encoder in collaborative filtering.
- VAE for Collaborative Filtering (Mult-VAE) [2].The method combines VAE with collaborative filtering recommendation system.
- DAE for Collaborative Filtering (Mult-DAE) [2]. It comes from the same article as Mult-VAE, and also uses multinomial likelihood, but here it uses denoising autoencoder structure.
- Collaborative denoising auto-encoders (CDAE) [21]. This is a cooperative scheme of denoising autoencoder. CDAE assumes that the iteration of user-item we observe is a corrupted version of the user's complete ground truth.
- Disentangled VAE for Recommendation (MacridVAE) [3]. This method argues that a user's behavior often involves the superposition of different intentions, so the user's intentions are separated from macro and micro directions, and only the intentions related to the current category are considered when predicting a certain category.
- A new variational autoencoder structure for recommendation (RecVAE) [22]. RecVAE is a new method to set super parameters for β-VAE framework, which can training model based on alternate updating.

E. Experimental Results and Analysis

In this section, we quantitatively compare our proposed methods with various autoencoder-based baselines. Among them, there are many VAE-based methods that serve the Top-K task in the original paper. Here, we modify them to evaluate the score prediction, so the performance may be lost. Table II report the evaluation results on the Movielens, Douban, and Flixster data sets.

Taking the RMSE index as an example, on the three data sets, MixVAE increased by 4.3% on average compared with I-AutoRec, 5.4% on average compared with CDAE, and 3.6% on average compared with RecVAE. However, Mult-VAE and Mult-DAE, which have achieved better results in the field of ranking, are far behind in score prediction, 10.6% and 11.8% weaker than our method, respectively. MacridVAE, which is

also a sorting algorithm, has poor experimental results and even lags far behind other baseline algorithms. A possible conjecture is that MacridVAE's multi-level de-entanglement is more inclined to the comparison of priorities between items, so it can have a better effect when sorting, but it is not ideal when predicting the specific rating of preference.

We can observe that MixVAE performs better than the same VAE-based models RecVAE and Mult-VAE, which indicates that our model is very effective in relieving the posterior collapse of VAE by adding a deterministic autoencoder. On the other hand, we know that the noise of the DAE model is added to the front-end data, while the VAE model can be regarded as adding noise to the hidden vector of the middle layer. In the experiment, the CDAE model based on the denoise scheme is better than the model based on VAE, that is, the front-end noise is better than the middle layer noise in the rating prediction task. And our method once again surpasses the scheme based on the denoising autoencoder, which indicates that VAE can benefit from the deterministic transformation of the encoder.

F. Ablation Experiment

In order to verify the effectiveness of the module, we carried out ablation experiments. We tested two branches of the whole model separately, and the performance test results of the separate parts are shown in table III. It can be seen that compared with the two independent parts, the whole model is improved by 4.2% and 10.3%, which proves that our model is effective.

TABLE III
RMSE OF ABLATION EXPERIMENT.

Model	MovieLens	Douban	Flixster	Average Promotion
DeepAE	0.845	0.750	0.933	+4.2%
VAE	0.860	0.873	0.929	+10.3%

VI. CONCLUSION AND FEATURE WORK

In this paper, the mismatch between encoder and decoder used in the VAE recommendation system is discussed, which is related to the posterior collapse phenomenon. We regard the incompatibility between encoder and decoder as the poor mapping from data manifold to parameter space, and then improve the posterior network to make up for the incompatibility and ease posterior collapse. We extend VAEs with a deterministic encoder to balance coding and decoding. Experiments on general recommendation data show that our method alleviates the posterior collapse in VAE and improves the recommendation performance.

ACKNOWLEDGMENT

This work is supported by Tianjin "Project + Team" Key Training Project under Grant No. XC202022, the National Nature Science Foundation of China under Grant No. 61702368, and the Natural Science Foundation of Tianjin under Grant No. 18JCQNJC00700.

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[2] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 689–698.
[3] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," *arXiv preprint arXiv:1910.14238*, 2019.
[4] C.-C. Lin, Y. Hung, R. Feris, and L. He, "Video instance segmentation tracking with a modified vae architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 147–13 157.
[5] O. Rybkin, K. Daniilidis, and S. Levine, "Simple and effective vae training with calibrated decoders," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9179–9189.
[6] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *International conference on machine learning*. PMLR, 2017, pp. 3881–3890.
[7] P. Z. Wang and W. Y. Wang, "Riemannian normalizing flow on variational wasserstein autoencoder for text modeling," *arXiv preprint arXiv:1904.02399*, 2019.
[8] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," *arXiv preprint arXiv:1804.00891*, 2018.
[9] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," *arXiv preprint arXiv:1901.05534*, 2019.
[10] C. Wu, P. Z. Wang, and W. Y. Wang, "On the encoder-decoder incompatibility in variational text modeling and beyond," *arXiv preprint arXiv:2004.09189*, 2020.
[11] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.
[12] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
[13] E. Nalisnick, L. Hertel, and P. Smyth, "Approximate inference for deep latent gaussian mixtures," in *NIPS Workshop on Bayesian Deep Learning*, vol. 2, 2016, p. 131.
[14] G. B. Dantzig, A. Orden, P. Wolfe *et al.*, "The generalized simplex method for minimizing a linear form under linear inequality restraints," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 183–195, 1955.
[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[16] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in $beta$-vae," *arXiv preprint arXiv:1804.03599*, 2018.
[17] O. Kuchaiev and B. Ginsburg, "Training deep autoencoders for collaborative filtering," *arXiv preprint arXiv:1708.01715*, 2017.
[18] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.
[19] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," *arXiv preprint arXiv:1704.06803*, 2017.
[20] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, K. Li, Y. Chen, Y. Lu, H. Wang, C. Tian, X. Pan *et al.*, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," *arXiv preprint arXiv:2011.01731*, 2020.
[21] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the ninth ACM international conference on web search and data mining*, 2016, pp. 153–162.
[22] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko, "Recvae: A new variational autoencoder for top-n recommendations with implicit feedback," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 528–536.