Vector-Quantized Autoencoder With Copula for Collaborative Filtering

Guanyu Wang
University of Electronic Science and Technology of China

Ting Zhong
University of Electronic Science and Technology of China

Xovee Xu
University of Electronic Science and Technology of China

Kunpeng Zhang
University of Maryland, College park

Fan Zhou*
University of Electronic Science and Technology of China
fan.zhou@uestc.edu.cn

Yong Wang
Zhengzhou Aiwen Computer Technology Co., Ltd., China.
The Hong Kong University of Science and Technology

ABSTRACT

In theory, the variational auto-encoder (VAE) is not suitable for recommendation tasks, although it has been successfully utilized for collaborative filtering (CF) models. In this paper, we propose a Gaussian Copula-Vector Quantized Autoencoder (GC-VQAE) model that differs prior arts in two key ways: (1) Gaussian Copula helps to model the dependencies among latent variables which are used to construct a more complex distribution compared with the mean-field theory; and (2) by incorporating a vector quantisation method into encoders our model can learn discrete representations which are consistent with the observed data rather than directly sampling from the simple Gaussian distributions. Our approach is able to circumvent the "posterior collapse" issue and break the prior constraint to improve the flexibility of latent vector encoding and learning ability. Empirically, GC-VQAE can significantly improve the recommendation performance compared to existing state-of-the-art methods.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender system, vector quantisation, Gaussian Copula, collaborative filtering, variational autoencoder

ACM Reference Format:
Guanyu Wang, Ting Zhong, Xovee Xu, Kunpeng Zhang, Fan Zhou, and Yong Wang. 2021. Vector-Quantized Autoencoder With Copula for Collaborative Filtering. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482216

*Corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482216

1 INTRODUCTION

The rapid development of the Internet has aroused the emergence of massive data. As an indispensable tool for information retrieval and alleviating information overloading, recommender system has become a research hotspot in the field of information and knowledge management. As one of the most commonly used methods on recommendation tasks, Collaborative Filtering (CF) selects the items that users are interested in according to the selection of groups with common preferences. In recent years, a few studies have applied neural networks to build more powerful CF models [1, 4, 15]. Variational auto-encoder (VAE) is considered a robust feature extraction tool and has been incorporated into CF models [3, 7, 8, 11, 14]. VAE-based CF models can extract the features of observed data as latent representations and recommend items by using an encoder-decoder architecture. However, they still face several notable challenges despite their early success.

Challenges. When dealing with recommendation tasks, earlier VAE-based CF models often: (1) face the problem of "posterior collapse", where the learned latent variables contain relatively little information and thus the model is unable to train an effective encoder, only strengthen the decoder's auto-regression ability; (2) specify the prior distribution of latent variables as static standard Gaussian distribution, which makes VAE unsuitable for recommendation tasks, i.e., the constraint of prior distribution limits the flexibility of encoders and weakens the recommendation performance.

Contributions. In this paper, we introduce a Gaussian Copula into the Quantized Autoencoder model for collaborative filtering and make the following contributions. First, our model uses two methods to circumvent the "posterior collapse" issue: (1) the Gaussian Copula for modeling the correlations of latent variables to obtain complex and accurate posterior distributions; and (2) the Quantized Autoencoder for learning discrete representations to mitigate the impact of noise data. Second, our model removes the restriction of prior distribution in the loss function during the training process, making the encoder more flexible and suitable for recommendation tasks. Experiments show that our method outperforms several state-of-the-art baselines.

2 METHODOLOGY

Figure 1 shows the overall architecture of our proposed GC-VQAE model. The encoder of GC-VQAE consists two neural networks and one embedding table which is used to convert the observation

data into discrete latent representations. The decoder network is to recover the input data in a generative process.

2.1 Correlation Construction

First, we use x and z to represent observed data and latent variables, respectively. In general, x is a sparse matrix that records the users' click history for each item and z is the dense latent representation of x. In typical VAE-based CF models, each variable z_i of z is assumed to be Gaussian distribution with mean μ_i and variance σ_i:

$$q_\phi(z_i|x) = \mathcal{N}(\mu_i, \sigma_i^2). \tag{1}$$

Most VAE-based models adopt the mean-field theory in which latent variables are independent from each other. Despite their advantages such as simple and effective, they often face several notable problems. As the latent variables to be simple, they cannot accurately represent the characteristics of the observed data. Also, the reparametric sampling operation introduces a large number of noise data which prevent the model from effectively utilizing the information of latent representations generated by the encoder, and in consequence the decoder can only continuously strengthen its autoregressive ability to improve the model performance. Such problems in VAE-based models are known as "posterior collapse".

Previous works have demonstrated that constructing more complex distributions for latent variables mitigates the "posterior collapse" issue effectively [9]. Motivated by prior success, we introduce Gaussian Copula into the variational inference for a further performance boost. Considering there exists complex correlations between latent variables. For example, as shown in Figure 2, we use two variables to represent four features. Each user likes two types of items: (1) the items with features A and B; and (2) the items with features C and D. If the model is based on the mean-field theory, it can only construct a simple posterior distribution and make inaccurate predictions. In contrast, if we build the correlations between latent variables, more accurate posterior distributions can be inferred, and better recommendations can be made.

According to the Sklar's theorem, for any multivariate joint distribution H of n random variables, we can use a Copula function to connect their respective marginal probability distributions $F_j(x) = P(X_j \leq x)$, such that:

$$H(x_1, x_2, ...x_n) = C(F_1(x_1), F_2(x_2), ..., F_n(x_n)). \tag{2}$$

Successfully constructing the distribution H means that we can capture the correlations between latent variables and obtain a posterior distribution with more information contained [12]. Compared with the posterior distribution whose variables are independent from each other, the latent representations we sampled from H are more effectively to represent the features of observation data. Here we introduce our method: augmenting the mean-field distribution with a copula.

For multivariate standard Gaussian distributions, the key to model their copula function is to construct the convariance matrix Σ between all the variables, the Copula function $GC(\cdot)$ is as follows:

$$GC(\Phi_1(x_1), \Phi_2(x_2), \cdots, \Phi_n(x_n); \Sigma) = \Phi_\Sigma(x_1, x_2, \cdots, x_n). \tag{3}$$

As depicted in Figure 1, we use two deep networks to model our method: inference network and correlation network. Inference network is used to generate independent Gaussian distributions for each latent variable $q_\phi(z_i|x) = \mathcal{N}(\mu_i, \sigma_i^2)$. As for correlation network, it constructs the convariance matrix Σ as follows:

$$\zeta = \tanh(W \cdot x + b) \tag{4}$$

$$\Sigma = I + \zeta \cdot \zeta^T. \tag{5}$$

The convariance matrix Σ constructed is positive definite and real symmetric, and we use it for later reparameterization in GC-VQAE.

2.2 Reparameterization in Copula-Augmented Distributions

As illustrated in Section 2.1, we propose to use Copula-augmented Gaussian distribution for latent variables z with respect to mean, variance of each variable's independent distribution, and their covariance matrix. When randomly sampling from the distribution we need to prevent the gradient vanishing problem when training the model, which requires an indirect random sampling strategy. To overcome this hurdle, we apply the reparameterization trick for multivariate joint distribution sampling proposed in a previous work [16].

Compare to the reparameterization trick of previous VAE models which are based on mean-field theory, our method first samples from the standard Gaussian distribution, and then transform the samples to ensure their values consistent with the values obtained in the original multivariate joint distribution. Recall that for users' observed data, we use M-dimensional vector z as latent variables. We obtain the independent marginal Gaussian distribution of each z_i and the convariance matrix Σ. Since Σ is positive definite according to Eq. (5), we can use Cholesky decomposition(a method for matrix decomposition): $\Sigma = AA^T$ to get matrix A. Then we sample vector

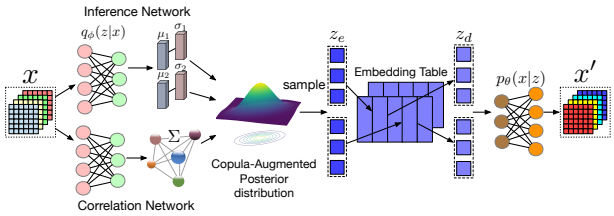

Figure 1: The overall architecture of GCVQAE

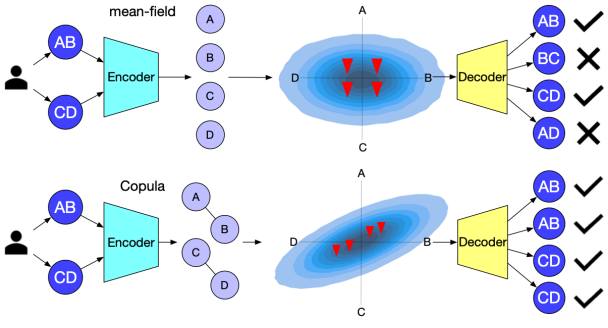

Figure 2: Two different theories of constructing the posterior distribution.

$\epsilon \sim \mathcal{N}(0, I)$ which has a dimension of M. We multiply the vector A and ϵ to have sample $\xi = A \cdot \epsilon$ such that $\xi \sim \mathcal{N}(0, \Sigma)$. Therefore, the obtained ξ is equivalent to the values sampled from multivariate standard normal distribution with the same covariance matrix Σ. We then convert the ξ to z via the following transformation:

$$z_i^{sample} = Q_\phi^{-1}(z_i|x)(\Phi(\frac{\xi_i}{\sigma_i})), \tag{6}$$

where $Q_\phi(z_i|x)$ and $\Phi(\cdot)$ are cumulative distribution function (CDF) of $q_\phi(z_i|x)$ and standard Gaussian, respectively. σ_i is the standard deviation of ξ_i. The samples obtained by GC-VQAE's reparameterization trick retain the information of constructed Copula-augmented posterior distributions, which can be used to decode and generate better recommendation items. The model's gradients can also be back-propagated through the sampled vector z.

2.3 Vector Quantisation for Latent Coding

In this section, we give the details of our proposed vector quantisation method, which aims to make our model more suitable for recommendation task. In traditional VAE-based models, $\mathcal{L}_{\text{ELBO}}$ is the optimization objective defined as follows:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(x)}\left[\log p_\theta(x|z) + \log \frac{p_\theta(z)}{q_\phi(z|x)}\right], \tag{7}$$

where $p(z)$ is regarded as the prior distribution of latent variables that cannot be calculated directly. Thus most VAE-based models consider $p(z)$ as a standard Gaussian distribution and let the constructed posterior distribution $q(z|x)$ close to the static prior distribution. This assumption works greatly in many generative scenarios as they can draw samples randomly from the standard Gaussian distribution and generate various results with a well trained decoder. However, recommendation models generally do not require such prior distribution, which, limits the flexibility of the decoder.

Therefore in GC-VQAE, we replace the sampled z with discrete codes and remove the restriction of prior distribution which renders more coding space for encoders. We use z_e to denote the sampled z in the following paragraphs.

First, we divide z_e into groups, the dimension of each group is K, so the number of groups is $n = \frac{M}{K}$. We build and maintain an embedding table E, $e \sim \mathbb{R}^{K \times D}$ is the latent embedding space. We use the following transformation to replace $z_e^k (k \in \{1, \cdots, n\})$ with its nearest embedding vector e_i:

$$z_d^k = e_i, \quad \text{where} \quad i = \text{argmin}_j ||z_e^k - e_j||_2. \tag{8}$$

Then we use the converted $z_d = (z_d^1, \cdots, z_d^n)^T$ as input to the decoder network. The optimization objective of the model includes two parts: (1) training the decoder $p(x|z_d)$ to make reconstructed data similar to the observed data; and (2) optimizing the encoder network to obtain effective latent representations. Since the gradients cannot back-propagate through the "argmin$(\cdot)$" function, we use the following optimization objective instead:

$$\mathcal{L}_{\text{loss}} = \log p(x|z_d) + \alpha \cdot ||\text{sg}[z_e] - e||_2^2 + \beta \cdot ||z_e - \text{sg}[e]||_2^2, \tag{9}$$

where sg stands for *stop calculating* the gradients of a term. To improve the performance of the encoder, the loss function makes z_e and e close to each other and controls their change via parameter α and β. We tend to reduce their distance by changing the value of e thus α is generally larger than β. With using vector quantisation method in latent coding, the model effectively circumvents "posterior collapse" issue and improves the flexibility of the encoder, which brings obvious advantages when recommending items.

Table 1: Descriptive statistics of three datasets.

Dataset	# users	# items	# rating	density
lastfm	1,693	16,410	82,989	0.299%
ML-1M	6,940	3,952	1,000,209	3.65%
Gowalla	29,858	40,981	1,027,370	0.084%

3 EXPERIMENT

3.1 Experiment Settings

Dataset. Our experiments are conducted on three benchmark datasets: lastfm, ML-1M, and Gowalla. The basic statistics of three datasets are summarized in Table 1.

Implementation details. We implement our proposed GC-VQAE in Pytorch. We divide each dataset into training, validation, and test sets according to the scale of 8:1:1. The learning rate is 0.001 and we train the model with the Adam optimizer [6]. Hyper-parameters α and β are set to 0.4 and 0.2, respectively. The evaluation metrics are NDCG@20, NDCG@100, and Recall@50.

Baselines. We compare the performance of our proposed GC-VQAE with the following seven methods:

- **WMF [5]:** is a classic matrix factorization method.
- **SLIM [10]:** is a linear model which learns an asymmetric item-similarity matrix.
- **NeuMF [4]:** a matrix factorization model which explores the nonlinear interaction between user and item representations.
- **CMN [2]:** a memory-based CF model.
- **NGCF [13]:** is a graph-based CF method which integrates the user-item interactions.
- **CDAE [15]:** is an augmented denoising autoencoder.
- **Mult-VAE [8]:** is a state-of-the-art VAE-based CF model.

3.2 Experimental Results

Table 2 present the performance of our proposed GC-VQAE and other baseline models in terms of NDCG and Recall. Notably, we can see that limited by the modeling ability of linear model, WMF and SLIM are not competitive. Compared to linear models, the neural network-based models can acquire better performance over most of metrics. Mult-VAE outperforms other baselines, mainly because it uses KL-annealing to weaken the constraints brought by the static prior distribution in VAE. Although Mult-VAE effectively utilizes the feature extraction and autoregressive ability of VAE, it does not solve the problems in applying VAE to the recommendation model. As for GC-VQAE, it not only breaks the constraint of static prior distribution by using vector quantisation, but also models the complex correlations between latent variables and leads to a better approximation of the true posterior distribution by introducing Gaussian Copula into VAE-based CF models. GC-VQAE effectively addresses the "posterior collapse" problem and achieves significantly recommendation performance improvements.

Parameter sensitivity. To verify the effect of factor K on the performance of our model, we show model's performance at different

Table 2: Performance comparison on three datasets.

Dataset	lastfm			ML-1M			Gowalla		
Metric	NDCG@20	NDCG@100	Recall@50	NDCG@20	NDCG@100	Recall@50	NDCG@20	NDCG@100	Recall@50
WMF	0.197	0.268	0.313	0.268	0.355	0.411	0.095	0.159	0.230
SLIM	0.203	0.276	0.315	0.285	0.364	0.429	0.115	0.172	0.257
NeuMF	0.207	0.280	0.324	0.273	0.362	0.426	0.113	0.168	0.253
CMN	0.211	0.285	0.321	0.278	0.384	0.434	0.116	0.172	0.255
CDAE	0.217	0.291	0.327	0.287	0.374	0.435	0.121	0.179	0.269
NGCF	0.217	0.296	0.334	0.298	0.381	0.439	0.123	0.182	0.267
Mult-VAE	0.219	0.295	0.339	0.299	0.395	0.452	0.125	0.186	0.278
GC-VQAE	**0.233**	**0.306**	**0.357**	**0.321**	**0.407**	**0.457**	**0.125**	**0.187**	**0.286**

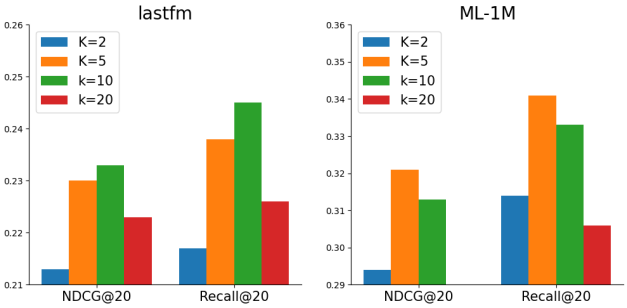

Figure 3: Parameter sensitivity of GC-VQAE.

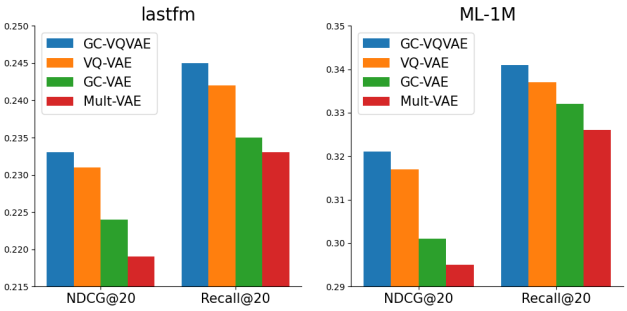

Figure 4: Ablation study on lastfm and ML-1M datasets.

values of K in Figure 3. The result suggests that our GC-VQAE model achieves best performance with 10 on lastfm and $K = 5$ on ML-1M. When K changes, the final performance varies greatly. This phenomenon demonstrates that the dimension of an embedding group has a vital influence on the recommendation performance.

Ablation study. We perform ablation study by separately showing how two essential components of GC-VQAE (Gaussian copula and vector quantisation) affect its recommendation performance. We create two variants of GC-VQAE by including the Gaussian copula or vector quantisation part into VAE model, resulting in GC-VAE and VQ-VAE, respectively. The experimental results are shown in Figure 4, which proves that both components contribute to the performance improvement and vector quantisation improves larger than Gaussian copula.

To investigate the effectiveness of encoding initial latent representation z_e, we implement another variant GC-VQAE*, which

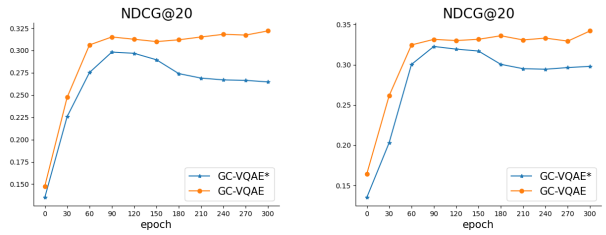

Figure 5: Convergence of GC-VQAE and GC-VQAE*

uses MLP to generate z_e directly instead of drawing samples from distributions. We present the training procedure of GC-VQAE and GC-VQAE* on ML-1M dataset in Figure 5. The result shows that the performance of GC-VQAE is better than GC-VQAE*. With the training time increases, GC-VQAE* tends to overfitting to the data. We speculate that when the latent factors are presented in the form of distributions, noise data will be introduced into the model during the sampling process, which enhances the robustness of the model and prevents the overfitting phenomenon. In addition, Gaussian Copula and vector quantisation effectively solve the "posterior collapse" problem which may be caused by these noise data in vanilla VAE model and improves the recommendation performance.

4 CONCLUSION

In this work, we argued the improper design of VAEs for collaborative filtering. We proposed GC-VQAE that leverages a Gaussian Copula and vector quantisation to address the "posterior collapse" issue. Our GC-VQAE model can capture the complex correlations between latent variables and make the constructed distribution closer to the true posterior distribution. The sampled latent representations are replaced with discrete embedded vectors, which enables the model to make better use of the information in the latent representations. The optimization objective does not contain the restriction of the prior, which gives more coding space for the encoders. Extensive experiments showed that our model outperforms several state-of-the-art baselines.

5 ACKNOWLEDGMENTS

Thiswork was supported by National Natural Science Foundation of China (Grant No.6217021634, No.62072077 and No.62102326).

REFERENCES

[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*. 191–198.

[2] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *The International SIGIR Conference on Research & Development in Information Retrieval*. 515–524.

[3] Ehtsham Elahi, Wei Wang, Dave Ray, Aish Fenton, and Tony Jebara. 2019. Variational low rank multinomials for collaborative filtering with side-information. In *Proceedings of the ACM Conference on Recommender Systems (RecSys)*. 340–347.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the International Conference on World Wide Web (WWW)*. 173–182.

[5] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining (ICDM)*. IEEE, 263–272.

[6] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[7] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*. 305–314.

[8] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the world wide web conference (WWW)*. 689–698.

[9] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. 2019. Don't blame the Elbo! a linear Vae perspective on posterior collapse. *arXiv preprint arXiv:1911.02469* (2019).

[10] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *IEEE International Conference on Data Mining (ICDM)*. IEEE, 497–506.

[11] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential variational autoencoders for collaborative filtering. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*. 600–608.

[12] Dustin Tran, David M Blei, and Edoardo M Airoldi. 2015. Copula variational inference. *arXiv preprint arXiv:1506.03159* (2015).

[13] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the International SIGIR conference on Research and development in Information Retrieval*. 165–174.

[14] Zhitao Wang, Chengyao Chen, Ke Zhang, Yu Lei, and Wenjie Li. 2018. Variational Recurrent Model for Session-based Recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. 1839–1842.

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

[16] Ting Zhong, Guanyu Wang, Joojo Walker, Kunpeng Zhang, and Fan Zhou. 2021. Variational Autoencoder with Copula for Collaborative Filtering. In *Workshop on Deep Learning Practice for High-Dimensional Sparse Data with KDD 2021(DLP-KDD)*.