Variational Autoencoder with Copula for Collaborative Filtering

Ting Zhong
University of Electronic Science and Technology of China
Chengdu, China

Guanyu Wang
University of Electronic Science and Technology of China
Chengdu, China

Joojo Walker
University of Electronic Science and Technology of China
Chengdu, China

Kunpeng Zhang
University of Maryland, College park
Maryland, United States

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

ABSTRACT

Variational autoencoder (VAE) has been successfully utilized for collaborative filtering (CF) models. Although effective, it still has some limitations that can be improved. For instance, there exist correlations between different features of the observation data. Accurately capturing these correlations is challenging and can significantly impact VAE-based CF models' performance. Moreover, earlier VAE-based CF models model the families of distributions based on the mean-field theory, which greatly limits their generalizability. This paper addresses this research gap by using Gaussian copula to help the variational model preserve dependency among latent variables. Besides, to make Gaussian Copula perform well in our VAE-CF model, we design a novel reparameterization technique in the sampling process. Consequently, our approach is able to construct more complex distributions, and the resulting variational family can better approximate the posterior distributions. Finally, we empirically demonstrate the superiority of our proposed method over several state-of-the-art baselines on three real-world datasets.

KEYWORDS

Recommender system, VAE, Gaussian Copula, collaborative filtering, reparameterization

ACM Reference Format:
Ting Zhong, Guanyu Wang, Joojo Walker, Kunpeng Zhang, and Fan Zhou. 2018. Variational Autoencoder with Copula for Collaborative Filtering. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The large volume of user–item interaction data has facilitated the design of several personalized recommendation models with the aim of presenting to users a set of unseen items they may like. Collaborative filtering (CF) is one of the most commonly used technologies in recommender systems. In recent years, due to the strong modeling ability of neural networks, a lot of studies have incorporated neural networks into their design of powerful collaborative filtering algorithms [4, 13] and achieved improved performance. Previous works have incorporated variational autoencoder (VAE) in collaborative filtering by extending the linear latent factor models to non-linear probabilistic models via neural networks. VAE-based CF models [9, 12] have been evaluated on large-scale datasets and demonstrated promising results.

Challenges: To make the inference tractable, many VAE-based CF models employ the mean-field theory [3, 7], where each latent variable is regarded as independent. This assumption enables efficient variational inference but sacrifices accuracy to a certain extent. In many cases, there exist dependencies over the latent variables of VAE, but the mean-field VAE models fail to capture such correlations. Understanding the correlations between latent variables is critical and can significantly improve the performance of the recommendation models.

Contributions: In this paper, we address this issue by introducing Gaussian Copula into the VAE model for collaborative filtering. As a classical method in statistics, the copula method can separate the correlation between random variables from the marginal distribution of variables. Based on the copula theory, we model the dependencies of latent variables via a covariance matrix to obtain a more intricate multivariate distribution that can better approximate the real posterior distribution. In order to make the Gaussian Copula VAE model perform well in collaborative filtering, we design a novel reparameterization technique in the sampling process of a copula family. Experiments on three real-world datasets show that our method outperforms several cutting-edge baselines.

2 METHODOLOGY

The overall architecture of our Gaussian Copula Variational Autoencoder **(GCVAE)** model is shown in Figure 1. GCVAE consists of two main components: the variational gaussian copula inference (VGCI) and the generative network. VGCI has two networks parameterized by ϕ and η. In particular, the inference network is used for the usual variational inference and the correlation network is designed to capture the important latent correlations. The generative network parameterized by θ is used for the generative process.

DLP-KDD 2021, August 15, 2021, Singapore

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

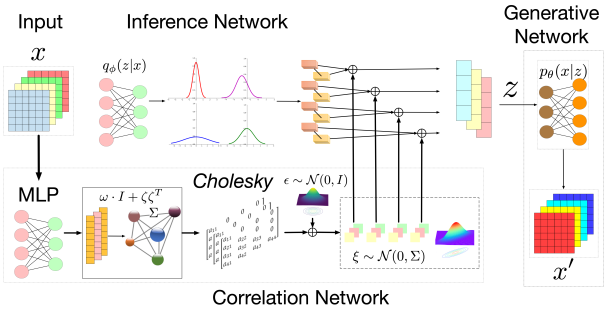


Figure 1: The overall architecture of Gaussian Copula VAE.

2.1 Variational Gaussian Copula Inference

Mapping user input data to appropriate latent variables is a crucial step in any generative model. The variational autoencoder set the distribution of potential variable to be a Gaussian distribution.

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

where z represents the multiple variables, $\mathcal{L}_{ELBO}$ represents the evidence lower bound. The real posterior distribution p is fixed, so we need to adjust parameters to maximize $\mathcal{L}_{ELBO}$ through minimizing the KL divergence.

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(x)}\left[\log p_\theta(x|z) + \log \frac{p_\theta(z)}{q_\phi(z|x)}\right]. \tag{2}$$

In the mean-field theory, each latent variable is regarded as independent. Assume that the latent variable z has k dimensions, it calculates $q_\phi(z|x) = \prod_{j=1}^{k} q_j(z_j)$. For scalable variational inference, we consider that latent variables are not independent of each other in the recommendation model. At this point, the Copula method is an appropriate tool to model the correlation of random variables with known marginal probability distributions. We use the variational gaussian copula inference to approximate the true posterior distribution [10].

According to Sklar's theorem, any multivariate joint distribution H of n random variables can be decomposed into respective marginal probability distributions $F_j(x) = P(X_j \leq x)$ and a Copula function, such that

$$H(x_1, x_2, ...x_n) = C(F_1(x_1), F_2(x_2), ..., F_n(x_n)). \tag{3}$$

In this form, the randomness and coupling of the variables are separated. The marginal probability distribution of each variable reflects the randomness of variables, and the copula function aims to capture the coupling between variables. In other words, the nature of a joint distribution with respect to correlation is entirely determined by its copula function.

Therefore, as long as we have a copula function, we can flexibly construct their joint distribution and calculate the joint probability density function via derivation. Let $c(\cdot)$ represent the copula density function.

$$f(x_1, x_2, ..., x_n) = c(F_1(x_1), F_2(x_2), ..., F_n(x_n)) \prod_{i=1}^{n} f_i(x_i). \tag{4}$$

We can get the copula-augmented variational family:

$$q(z|x;\phi,\eta) = c\left(Q_1\left(z_1;\phi\right), \ldots Q_k\left(z_k;\phi\right);\eta\right) \prod_{i}^{k} q\left(z_i|x;\phi\right). \tag{5}$$

We substitute $q(z|x;\phi,\eta)$ in Eq.(5) for $q_\phi(z|x)$ in Eq. (2) to better approximate the true posterior distribution.

To model our novel VGCI method, as depicted in Figure 1, we first analyze the input data through an inference network and a correlation network. The inference network is used to generate the gaussian distributions parametrized by ϕ with various mean μ_i and variance σ_i for each latent variable z_i:$q_\phi(z_i|x) = (\mu_i, \sigma_i^2)$. The correlation network is designed to generate the covariance matrix of their joint distribution parametrized by η. In order to make the constructed covariance matrix Σ_η positive semidefinite and real symmetric, we first make the following transformations:$\Sigma_\eta = I + \zeta\zeta^T$. where ζ is a vector with the same dimension as the latent variable generated by the correlation network. I is an identity matrix. The constructed marginal distributions and the covariance matrix are combined to produce the approximated posterior distribution $q(z|x;\phi,\eta)$.

2.2 Conditioned Reparameterization and Sampling

As illustrated above, we construct the joint distribution of latent variables z with respect to the mean, the variance of independent distribution of each variable and their covariance matrix. In practice, it is difficult to sample from a multivariate joint distribution where there exist interactions between variables. To overcome this problem, we design a new sampling method from multivariate joint distributions, which allows our model to better reparameterize the latent variables and approximate the true posterior.

Suppose the dimension of latent variables is K. First, we can obtain matrix A by the Cholesky decomposition of the covariance matrix Σ_η such that $\Sigma_\eta = AA^T$. Then we sample $\epsilon \sim \mathcal{N}(0, I)$ being K-vectors. By multiplication, we can make a sample $\xi = A \cdot \epsilon$ where $\xi \sim \mathcal{N}(0, \Sigma)$. Note the ξ and z have the same covariance matrix, so we can convert ξ to z via the following transformation:

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

where σ_i is the standard deviation of ξ_i, $Q_\phi(z_i|x)$ is the CDF of $q_\phi(z_i|x)$ and $\Phi(\cdot)$ is the CDF of the standard Gaussian. Compared to the reparameterization trick used in VAE and previous generative recommendation models, our sampling strategy not only incorporates the uncertainty of user interest, but also accounts for the correlations of latent variables. Therefore, rather than relying on mean-field to sample independent variables, our method is encouraged to learn the relations between different attributes through the conditioned posterior of the variables. In the process of producing z, we use above reparametrization trick to ensure that the stochasticity in the sampling process is isolated and the gradient with respect to ϕ and η can be back-propagated through the sampled z and ξ.

2.3 Parameter Estimation

We can use the sampled z to obtain an unbiased estimate of $\mathcal{L}_{\text{ELBO}}$ and optimize it with stochastic gradient ascent. The $\mathcal{L}_{\text{ELBO}}$ is as follows:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} &= \mathbb{E}_{q_\phi(x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{q_\phi(x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z)}\right] \\ &= \mathbb{E}_{q_\phi(x)}\left[\log p_\theta(x|z)\right] - \mathbb{E}_{q_\phi(x)}\left[\log \frac{\prod_i^n q_\phi(z_i|x)}{p_\theta(z)}\right] \\ &\quad - \mathbb{E}_{q_\phi(x)}\left[-\frac{1}{2}\log|\Sigma_\eta| + \frac{1}{2}\xi^T(I-\Sigma_\eta^{-1})\xi\right],\end{aligned} \tag{7}$$

where the first term is the reconstruction cost, and the second and third terms are the KL divergence for typical VAE and copula, respectively. From Eq.(7) we can observe that $\mathcal{L}_{\text{ELBO}}$ is a function of parameters ϕ, θ and η. As the true distribution $\mathbb{E}_{q_\phi(x)}[\log p_\theta(x|z)]$ is difficult to calculate, we use Monte Carlo method for gradient ascent of this term. The training procedure is summarized in Algorithm 1.

Algorithm 1 The training of Copula-VAE.

Input: Users-item interaction matrix $X \in \mathcal{R}^{U\times I}$

Randomly initialize θ, ϕ, η
while not converged **do**
  Sample a batch of users $\mathcal{U}$
  for all user $u \in \mathcal{U}$ **do**
    Sample ξ and z_u from the generative process
    Compute noisy gradient $\nabla_\theta\mathcal{L}$, $\nabla_\phi\mathcal{L}$ and $\nabla_\eta\mathcal{L}$ with ξ ,z_u
  end for
  Average noisy gradients from batch
  Update θ, ϕ and η via stochastic gradient ascent
end while
return θ, ϕ and η

In practice, the kl-annealing factor is usually used to control the strength of regularization:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(x)}\left[\log p_\theta(x|z)\right] - \lambda\cdot(\text{KL}_1 + \text{KL}_2), \tag{8}$$

where KL_1 and KL_2 are the second and third terms of Eq.(7), i.e., the marginal KL divergence in typical VAE and the copula KL, respectively. Here, we use kl-annealing [1] to train our model starting with $\lambda = 0$, and gradually increasing to a certain threshold. We propose using $\lambda \neq 1$, which means that we are no longer optimizing a lower bound on the log marginal likelihood. If $\lambda < 1$, we are also weakening the influence of the prior constraint $p(z) = \mathcal{N}(z; 0, I_k)$.
Recommendation: Given a user's click history x, we can sample z from the generated multivariate joint distribution of latent variables using our trained model. The decoder transforms sampled z into scores of each item to this user. In a typical top-K recommendation system, we take the top-K value as the prediction items for this user.

3 EXPERIMENT

Dataset: We conduct our experiments on three datasets: MovieLens-100K (ML-100K), Movielens-1M (ML-1M), and Gowalla. Table 1 summaries the statistics of the datasets.

Table 1: Descriptive statistics of datasets.

Dataset	#users	#items	#rating	density
ML-100K	943	1682	100000	6.3%
ML-1M	6940	3952	1000209	3.65%
Gowalla	29858	40981	1027370	0.084%

Implementation Details: To be consistent with the previous models, we adopt the same architecture in our model. The overall architecture is $[I \to 600 \to 50 \to 600 \to I]$. We also utilize the data pre-processing and model evaluation techniques specified in previous work [7]. We set the learning rate of our model α to 0.001 and train it with Adam [6]. We use NDCG@20, NDCG@100, Recall@50 as metrics to evaluate the performance of all models.
Baseline Models: We compare the performance of our model with the following baselines:

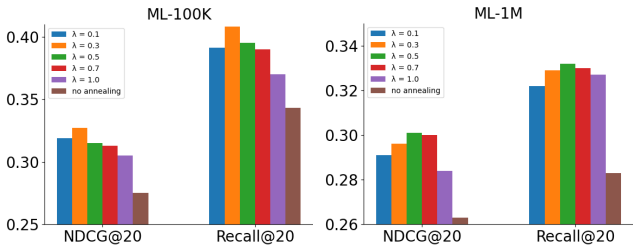
- **WMF [5]:** A matrix factorization method for item prediction from implicit feedback.
- **SLIM [8]:** A linear model which learns a sparse aggregation coefficient matrix by solving an ℓ_1-norm and ℓ_2-norm regularized optimization problem.
- **NeuMF [4]:** It uses a neural network to capture the nonlinear feature interactions of user and item embeddings.
- **CMN [2]:** A memory-based model learns a user-item specific neighborhood by encoding complicated user-item relations with the neural attention mechanism.
- **NGCF [11]:** This is a graph-based CF method, mainly following the standard GCN, including the use of nonlinear activation and feature transformation.
- **CDAE [5]:** An augmented standard denoising autoencoder which adds the latent factor to the input data.
- **Mult-VAE [7]:** A VAE-based CF model that introduces a different regularization parameter in the learning objective and tunes parameters using annealing.

Experimental Results: Table 2 summarizes the performance comparisons between our method and baseline approaches, which demonstrates that our model GCVAE achieves the best performance across all the datasets in terms of all metrics. Notably, we can see that linear models (WMF and SLIM) are not competitive. Compared to linear models, the neural network-based models exhibit better performance. Mult-VAE generally outperforms other baselines, but the improvement is insignificant. For example, CDAE, as a deterministic autoencoder-based approach, achieves comparable performance with Mult-VAE, which suggests that the vanilla VAE approach may not fully capture the true user-item interactions. The main reason lies in the independent variable modeling in Mult-VAE, which ignores the dependence between variables that may facilitate the user and item representation learning and thus the recommendation performance. This assumption can be proved by the result that our GCVAE achieves significant improvement over Mult-VAE. Clearly, this performance gain is attributed to GCVAE's ability of capturing salient correlations among latent variables, which leads to a better approximation of the true posterior distribution.

To investigate the effectiveness of the reparameterization trick that we tailored for the generative recommendation, we implement a variant of GCVAE, called GCVAE*, which replaces the sampling process in GCVAE with the reparameterization used in generative tasks. The fact that GCVAE* generally outperforms Mult-VAE proves the effectiveness of Gaussian copula. However, merely modeling the latent variables with copula still shows the inferior performance of GCVAE* compared to GCVAE. The performance gain of GCVAE not only proves the effectiveness of our new sampling

Table 2: Performance comparison on three datasets.

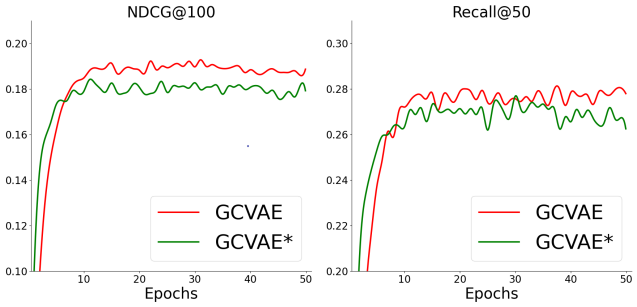
	ML-100K			ML-1M			Gowalla		
	NDCG@20	NDCG@100	Recall@50	NDCG@20	NDCG@100	Recall@50	NDCG@20	NDCG@100	Recall@50
WMF	0.283	0.389	0.530	0.266	0.353	0.411	0.092	0.157	0.231
SLIM	0.292	0.401	0.528	0.281	0.364	0.424	0.114	0.169	0.256
NeuMF	0.295	0.402	0.543	0.277	0.364	0.427	0.113	0.169	0.253
CMN	0.297	0.411	0.546	0.278	0.385	0.434	0.114	0.171	0.255
CDAE	0.305	0.419	0.557	0.288	0.376	0.434	0.121	0.178	0.267
NGCF	0.310	0.425	0.568	0.298	0.381	0.439	0.123	0.182	0.267
Mult-VAE	0.311	0.427	0.572	0.295	0.387	0.446	0.124	0.183	0.278
GCVAE*	0.315	0.432	0.576	0.299	0.395	0.450	0.122	0.180	0.271
GCVAE	**0.327**	**0.437**	**0.590**	**0.301**	**0.398**	**0.458**	**0.129**	**0.187**	**0.283**



(a) ML-100K **(b) ML-1M**

Figure 2: Parameter sensitivity of GCVAE.

method that may largely help the model generate better recommendation results, but also suggests that the generative recommender systems can learn better representations through approximating real posterior distributions rather than relying on simple Gaussian assumption of latent variables.



(a) NDCG@100 **(b) Recall@50**

Figure 3: Convergence of GCVAE and GCVAE*

Finally, we study the influence of kl-annealing factor λ, which is an important parameter in VAE-based recommender systems. We first conduct a grid search of the best value of λ and plot the results in Figure 2, which suggests that GCVAE achieves the best performance with a small value of λ, i.e., 0.3 on ML-100K and 0.5 on ML-1M. In addition, we present the training procedure of our model on Gowalla dataset in Figure 3, which demonstrates that our models (GCVAE and GCVAE*) can quickly converge to the best performance with a few epochs.

4 CONCLUSION

We have introduced copula into the VAE-based recommender framework, which enables us to capture the correlations between latent variables via a covariance matrix. The distribution of latent variables captured by our novel VGCI is close to the true posterior distribution. Subsequently, using the novel reparametrization technique we designed in the sampling process, we are able to generate better latent representations which are used to obtain better recommendation performance. Extensive experiments show that our model outperforms several state-of-the-art baselines.

REFERENCES

[1] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).
[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] 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.
[8] 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.
[9] 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.
[10] Dustin Tran, David M Blei, and Edoardo M Airoldi. 2015. Copula variational inference. *arXiv preprint arXiv:1506.03159* (2015).
[11] 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.
[12] 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.
[13] 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.