PRELIMINARY PREPRINT VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference.

Conditional Collaborative Filtering Process for Top-K Recommender System (Student Abstract)

Guanyu Wang, Xovee Xu, Ting Zhong*, Fan Zhou
University of Electronic Science and Technology of China
wgy05001@gmail.com, xovee@ieee.org, zhongting@uestc.edu.cn, fan.zhou@uestc.edu.cn

Abstract

Conditional neural process (CNP) has been extensively applied into data analyzing tasks due to its excellent ability to make accurate predictions for incomplete data points. However, in literature there are only few works that studied the CNP in recommendation systems. In this work, we propose CCFP, which is a collaborative filtering method that differs from other CF models by incorporating CNP into encoder-decoder architecture. By analyzing the complete user-item interaction data, our model fits a global representation that can better representing the features of users and items. CCFP can significantly improve the recommendation performance compared to baselines by predicting items for the target users with their incomplete observation data.

Introduction

The rapid development of the Internet has aroused massive user-item interaction data, making recommender systems the indispensable tool for information retrieval. As one of the most applied method in recommender systems, Collaborative Filtering (CF) has become a research hotspot in recent years. Among them, many works use variational auto-encoder (VAE) as the base model due to its strong learning ability for users/items. However, as a classical generative method, directly applying VAE into recommender systems faces several notable obstacles. For example, the constraint of prior distribution limits the flexibility of encoders and weakens the recommendation performance.

In this work, we incorporate conditional neural process (CNP) in designing collaborative filtering methods, which significantly improves the recommendation performance while retaining the encoder-decode architecture's efficiency.

Conditional neural process (Garnelo et al. 2018) aims to make accurate predictions for incomplete target data points after observing the complete training data points. CNP can flexibly obtain the distribution parameters of label y by analyzing the input data x rather than directly obtaining the label y.

In our proposed CCFP model, we first train a global representation by analyzing the user-item interaction data in training data and then use this representation to represent the features of overall data. Given the partial user-item interaction data, we combine them with the global representations to predict the recommendations for each user. Experiments conducted on two real-world datasets showed that our model outperforms strong baselines.

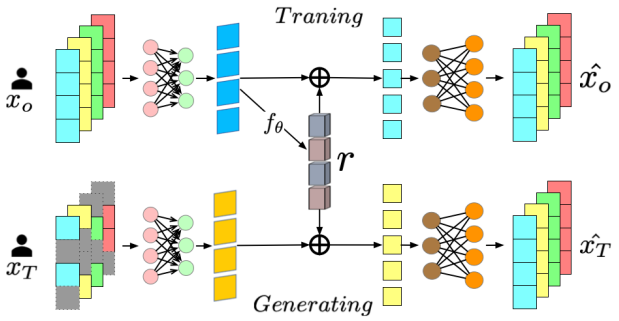

Figure 1: Overall architecture of CCFP

Method

We use $u_O \in \{u_1, u_2, \cdots, u_m\}$ to denote the observed users, $u_T \in \{u_{m+1}, u_{m+2}, \cdots, u_{m+n}\}$ to denote the target users and $i \in \{I_1, I_2, \cdots, I_k\}$ to denote items. Click matrix $X_O \in \mathbb{N}^{m \times k}$ and $X_T \in \mathbb{N}^{n \times k}$ are used to denote the user-item interaction matrices of u_O and u_T, respectively. The lower case x_O^u and x_T^u are vectors with the number of clicks for each item from u, respectively.

Model. Figure 1 shows the overall architecture of our proposed CCFP model. In general, there are a large number of items in a recommendation system, which makes the dimension of the click matrix very high. The first step of our model is to transform high-dimensional input data into low-dimensional latent factors by fully-connected layers. We use z_O^u (resp. z_T^u) to denote the latent representations of x_O^u (resp. x_T^u) that contains the features of the input data. The feature extraction method is similar to the encoder in VAE:

$$\begin{aligned} \mu, \sigma &= f_e(x) \\ z &= \mu + \sigma \cdot \epsilon, \epsilon \sim \mathcal{N}(0, I) \end{aligned} \tag{1}$$

In the training procedure of our model, we use obtained z_O^u to get the global representation r via neural network f_θ. In recommendation tasks, observation data usually account for

*Corresponding author
Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

a large proportion. Therefore, r can represent the features of the overall data to a great extent. We then concatenate z_O^u and r and generate the reconstructed vector $\hat{x_O^u}$ via network g_ϕ. Specifically, we use the following architecture:

$$\begin{aligned} r^u &= f_\theta(z_O^u) \\ r &= r^1 \oplus r^2 \oplus \cdots \oplus r^m \\ v^u &= concat(z_O^u, r) \\ \hat{x_O^u} &= g_\phi(v^u) \end{aligned} \tag{2}$$

In this way, we can get the reconstructed data $\hat{X}_O$. We expect to train $\hat{X}_O$ that is as close as possible to the original X_O and use the trained model to recommend items for users in the test set.

Training. Our model is trained by reducing the distance between the original data and the reconstructed data. The loss function is defined as follows:

$$\begin{aligned} \mathcal{L}_{loss} = -\frac{1}{m}[\sum_{u=1}^{m} x_O^u \cdot \log(\hat{x_O^u}) \\ + (1 - x_O^u) \cdot \log(1 - \hat{x_O^u})] \end{aligned} \tag{3}$$

Recommendation. Given a user's click history x_T^u, our model generates its latent representation z_T^u and concatenates it with r to get vector v^u, then g_ϕ will transform v^u 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.

Results and Conclusion

Datasets. We conduct our experiment on two real-world datasets: lastfm and Movielens-1M. The basic statistics of two datasets are summarized in Table 1.

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%

Table 1: Descriptive statistics of two datasets.

Baselines. We compare the performance of our proposed CCFP with the following baselines, both traditionally and state-of-the-art in recommendation systems: WMF (Hu, Koren, and Volinsky 2008), SLIM (Ning and Karypis 2011), NeuMF (He et al. 2017), NGCF (Wang et al. 2019), and Mult-VAE (Liang et al. 2018).

Experimental settings. Each dataset is split into training (80%), validation (10%), and test (10%) sets. The learning rate is 0.001 and we train the model with the Adam optimizer (Kingma and Ba 2014). The evaluation metrics are NDCG@20, NDCG@100, and Recall@50.

Results. Table 2 summarizes the performance comparisons between our method and baseline approaches, which demonstrates that our model achieves the best performance on both datasets and three metrics. Specifically, Mult-VAE outperforms other baselines which demonstrates the effectiveness of VAE's encoder-decoder architecture. In comparison to VAE-based recommendation methods, our model is more effective to deal with incomplete data points and utilize the information in observation data to generate more expressive user representations. As a result, we conclude that by incorporating CNP into collaborative filtering, the performance of recommendation can be improved.

Dataset	lastfm			ML-1M		
Metric	N20	N100	R50	N20	N100	R50
WMF	0.197	0.268	0.313	0.268	0.355	0.411
SLIM	0.203	0.276	0.315	0.285	0.364	0.429
NeuMF	0.207	0.280	0.324	0.273	0.362	0.426
NGCF	0.217	0.296	0.334	0.298	0.381	0.439
Mult-VAE	0.219	0.295	0.339	0.299	**0.395**	0.452
CCFP	**0.222**	**0.304**	**0.346**	**0.303**	**0.395**	**0.454**

Table 2: Performance comparison on three datasets.

Conclusion. We have introduced Conditional Neural Process into encoder-decoder based recommendation framework, which enables us to make more effective use of the information in the observation data to serve the target users. Compared to prior VAE-based CF models, we replace prior distribution based on empirical assumptions by the statistical information in the observation data and our model can avoid "posterior collapse" issue to some extent. Extensive experiments showed that our model outperforms several strong baselines.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62176043 and No. 62072077).

References

Garnelo, M.; Rosenbaum, D.; Maddison, C.; Ramalho, T.; Saxton, D.; Shanahan, M.; Teh, Y. W.; Rezende, D. J.; and Eslami, S. M. A. 2018. Conditional Neural Processes. In *ICML*, volume 80, 1690–1699.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW*, 173–182.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*, 263–272.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T. 2018. Variational autoencoders for collaborative filtering. In *WWW*, 689–698.

Ning, X.; and Karypis, G. 2011. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*, 497–506. IEEE.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *SIGIR*, 165–174.