Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder

Xu Zhao  
Tencent News  
Beijing, China  
xuzzzhao@tencent.com

Yi Ren  
Tencent News  
Beijing, China  
henrybjren@tencent.com

Ying Du  
Tencent News  
Beijing, China  
yingdu@tencent.com

Shenzheng Zhang  
Tencent News  
Beijing, China  
qjczyzhang@tencent.com

ACM Reference Format:
Xu Zhao, Yi Ren, Ying Du, Shenzheng Zhang, and Nian Wang. 2022. Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531902

1 INTRODUCTION

In the era of mobile internet, various online applications continuously emerge and explosively grow, in which recommendation systems play a key role in connecting users and content. Due to the excellent scalability and convenience of handling massive features, Embedding & MLP [4, 12, 13, 33] has become a paradigm for modern large-scale recommendation systems [5, 6, 23, 54, 55]. However, this paradigm is data demanding and suffers from cold-start problem [9, 17, 34]. Concretely, for a large amount of new emerging items with limited interactions, their embeddings are insufficiently trained which leads to poor recommendation performance. The cold-start problem has become a crucial obstacle for online recommendation. Under the influence of Pareto [31] effect in our industrial system, A small portion of well-trained items tend to obtain more accurate recommendations and more impressions, which will further compromise the distributing efficiency of system. Some approaches have been proposed to address the item cold-start challenge. CLCRec [46] proposes to address cold-start problem by maximizing two kinds of mutual information using contrastive learning technology. Heater [58] uses the sum squared error (SSE) loss to model the collaborative embedding. Notice that CLCRec and Heater are only designed for the CF-based backbone models. There are also some model-agnostic methods that could be widely equipped to various backbones. DropoutNet [39] applies dropout technology to relieve the model’s dependency on item ID. Meta Embedding [26] focuses on learning how to learn the ID embedding for new items with meta learning technology. MWUF [57] proposes meta Scaling and meta Shifting Networks to warm up cold ID embeddings. Although these methods are model-agnostic, they all have extra and strict requirements on data. For instance, DropoutNet and MWUF require the interacted user set of items for cold-start, while Meta Embedding requires building two mini-batches containing the same items for training. In the online scenario of industrial recommendation, these extra requirements on data stream make the deployment process rather difficult.

ABSTRACT

Embedding & MLP has become a paradigm for modern large-scale recommendation system. However, this paradigm suffers from the cold-start problem which will seriously compromise the ecological health of recommendation systems. This paper attempts to tackle the item cold-start problem by generating enhanced warmed-up ID embeddings for cold items with historical data and limited interaction records. From the aspect of industrial practice, we mainly focus on the following three points of item cold-start: 1) How to conduct cold-start without additional data requirements and make strategy easy to be deployed in online recommendation scenarios. 2) How to leverage both historical records and constantly emerging interaction data of new items. 3) How to model the relationship between item ID and side information stably from interaction data. To address these problems, we propose a model-agnostic Conditional Variational Autoencoder based Recommendation(CVAR) framework with some advantages including compatibility on various backbones, no extra requirements for data, utilization of both historical data and recent emerging interactions. CVAR uses latent variables to learn a distribution over item side information and generates desirable item ID embeddings using a conditional decoder. The proposed method is evaluated by extensive offline experiments on public datasets and online A/B tests on Tencent News recommendation platform, which further illustrate the advantages and robustness of CVAR.

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
Cold-Start Recommendation, Conditional Variational Autoencoder, Item ID embedding
Generally, there are two ways to solve the cold-start problem: the first is to mine the distribution patterns hidden in historical data [26, 43, 49, 53, 57], such as learning the transformation relationship between side information and item ID [26, 46, 57, 58]. The second is to improve the learning efficiency with limited samples of cold items, such as methods based on meta learning [19, 22, 25, 48]. However, previous works rarely consider both directions at the same time. In other words, methods in the first category only focus on the initialization of embedding, while the methods in the second category usually ignore patterns hidden in historical data.

Considering the issues mentioned above in previous research along with our industrial practice, we summarize three key points for the design of cold-start method: 1) How to conduct cold-start without additional data requirements and make strategy easy to be deployed in online recommendation scenario. 2) How the cold-start method can leverage both historical records and constantly emerging interaction data of new items. 3) How to model the relationship between item ID and side information stably from interaction data and minimize the discrepancy between cold item ID embedding and fully trained embedding space.

To achieve these desiderata, we propose a Conditional Variational Autoencoder based model-agnostic Recommendation (CVAR) framework. As an independent framework, CVAR could be equipped on various backbone models and be trained in an end-to-end way, using the same samples as backbone. Thus CVAR makes no redundant requirements for training data. In addition to giving desirable initialization in the cold-start phase, CVAR will leverage the continuously updating item ID embeddings from the backbone to generate enhanced warmed-up embeddings with superior quality. Therefore CVAR can fulfill the second concern mentioned above. As for the third problem mentioned before, CVAR aligns the representation of item ID and side information in the latent space of Conditional Variational AutoEncoder(CVAE) [36, 40, 50], which is shown in Figure 1. Specifically, previous works [26, 57, 58] usually train a learnable mapping from item side information to item ID. However, item ID contains not only content information, but also lots of interaction information which makes it difficult to learn a precise mapping directly. Inspired by Denoise Autoencoder [38], we first conduct dimension reduction of item ID embedding with Encoder-Decoder paradigm and get the denoised representation of item ID in a latent space. Then the side information is transformed to the latent space and aligned with the denoised representation of item ID. Corresponding with the design in CVAE, latent representation is defined as normal distribution which could maintain some Exploit-Expose ability for CVAR.

The main contributions of this work are summarized into four folds:

1. We propose a model-agnostic CVAR to warm up cold item ID embeddings. CVAR has no extra data requirements which makes it easy to be deployed in online scenario.
2. CVAR not only learns the pattern in historical data but also leverages the continuously updating item ID embedding from the backbone to generate enhanced warmed-up embeddings with superior quality.
3. We propose to model the relationship between item id and side information in the latent space and generate desirable ID embeddings using a conditional decoder.
4. Extensive offline and online experiments are conducted to demonstrate the effectiveness and compatibility of CVAR.

2 PROPOSED METHOD

In this section, we propose CVAR, a model-agnostic framework to warm up ID embeddings for new items. CVAR is designed based on the Click-Through-Rate(CTR) prediction task [10, 32], predicting the click/watch/purchase behavior in recommendation scenario, which is usually formulated as a supervised binary classification task. Each sample in CTR task consists of multiple input features \(x\) and the binary label \(y\). Generally, the input features \(x\) could be split into three parts, i.e. \(x = (i, X)\).

- \(i\), item ID, a unique number or string to identify each item in recommendation system.
- \(X = \{x_1, ..., x_{|X|}\}\), features which are used for CTR predict, may be categorical or continuous, such as the user attributes and contextual information.

Moreover, We take part of features from \(X\) as item side information \(I \subset X\), which will be consumed in item cold-start procedure. Standard feature preprocessing [10, 32] has been applied in this work. Continuous features are normalized to range between 0 and 1. Following the embedding technology [18, 24], categorical features are transformed to dense vectors, called embeddings. Normalized values of continuous features and dense embeddings of categorical features are concatenated together to constitute the final representation of input features. We denote the representations of \(i, I, X\) as \(v_i \in \mathbb{R}^d, v_f \in \mathbb{R}^{d|I|}\) and \(v_X \in \mathbb{R}^{d|X|}\) respectively. Note that \(v_f\) and \(v_X\) are the concatenation of multiple feature embeddings.

The CTR target is to approximate the probability \(\hat{y} = Pr(y = 1|x)\) by a discriminative function \(f(\cdot)\):

\[
\hat{y} = f(v_i, v_X; \theta) \tag{1}
\]

where \(\theta\) denotes the parameters of the backbone model \(f(\cdot)\). Then the Binary Cross Entropy [7] is used to format the loss function:

\[
\mathcal{L}(\theta, \phi) = -y \log \hat{y} - (1 - y) \log (1 - \hat{y}) \tag{2}
\]
where \( \phi \) denotes the parameters of the embedding layer, including \( \omega_i, \omega_f \) and \( \omega_X \). As the parameters is trained by historical data, the item ID embeddings of recent emerging items are rarely updated and stay around the initial point, leading to a low testing accuracy. It is known as the item cold-start problem.

As the relationship between cold-start and warm-up in recommendation system is confusing, we will give a brief discussion here. Generally speaking, cold-start strategies are applied to items which are known as almost cold items in Equation (6) or updating embeddings or distributions in autoencoder’s latent space is a generally used technology which has been proven effective in various fields [8]. Thus we consider mapping the side information embedding \( v_f \) into the same latent space and aligning with \( N_i(\mu, \Sigma) \). Specifically, Prior Encoder \( g'_\text{enc} \) maps \( v_f \) to \( N_i(\mu', \Sigma') \) and a wasserstein loss [27, 35] function is applied to aligning distribution \( N_i(\mu, \Sigma) \) with \( N_i(\mu', \Sigma') \):

\[
\mu', \sigma' = g'_\text{enc}(v_f; \omega'_\text{enc}); \mu' \in \mathbb{R}^k, \sigma' \in \mathbb{R}^k
\]

\[
z' \sim N_i'(\mu', \Sigma'); \Sigma' \in \mathbb{R}^{k \times k}, \text{trace}(\Sigma') = \sigma'
\]

\[
\hat{v}_i = g_\text{dec}(z', x_{\text{freq}}; \omega_\text{dec}); \hat{v}_i \in \mathbb{R}^d
\]

\[
L_w(w_\text{enc}, w'_\text{enc}) = W_2(N_i(\mu, \Sigma), N_i'(\mu', \Sigma'))
\]

where \( w'_\text{enc} \) denotes the parameters of \( g'_\text{enc} \), \( W_2(\cdot, \cdot) \) denotes the Wasserstein Distance. Instead of KL Divergence [2] or other distribution measurement, we choose Wasserstein Distance considering its symmetry and stability which is widely used in various scenarios [20, 45, 51, 52].

We will give an additional discussion about the design of \( x_{\text{freq}} \). Since it’s not feasible to extract frequency information from side information by \( g'_\text{enc} \) in item cold start scenario, the latent space ideally should contain no frequency information. However, the frequency information is already utilized in the origin item ID embedding \( v_i \) and automatically compressed to the latent space under the Encoder-Decoder framework. Thus we set \( x_{\text{freq}} \) as the a independent condition of decoder to reduce the proportion of frequency information in latent space.

In addition to \( L_{\text{rec}} \) and \( L_w \), we replace \( v_i \) in Equation (1) with enhanced warmed-up \( \hat{v}_i \) as item ID input to backbone and get the CTR loss \( L_{\text{ctr}} \) by forward computation:

\[
\hat{y}_{\text{warm}} = f(\hat{v}_i, \omega_X; \theta)
\]

\[
L_{\text{ctr}}(w_\text{enc}, w_\text{dec}) = -y \log \hat{y}_{\text{warm}} - (1 - y) \log (1 - \hat{y}_{\text{warm}})
\]

To avoid disturbing the recommendation of hot items, CVAE is taken as an independent module with backbone. During the training of CVAE, optimization of \( L_{\text{ctr}} \) in Equation (12) is only applied to \( w_\text{enc} \) and \( w_\text{dec} \), while the parameters \( \theta \) of \( f(\cdot) \) are fixed. We finally get the loss function to train CVAE:

\[
L_{\text{warm}}(w_\text{enc}, w_\text{dec}, w'_\text{enc}) = L_{\text{ctr}} + \alpha L_{\text{rec}} + \beta L_{\text{ctr}}
\]

where \( \alpha \) and \( \beta \) are hyperparameters to fuse the losses. Moreover, training of CVAE is along with the backbone’s training. As shown in Equation (13), CVAE will be trained by optimizing \( w_\text{enc}, w_\text{dec}, w'_\text{enc} \) to minimize \( L_{\text{warm}} \), using the same samples as training backbone, without any additional requirements on data. For a coming batch of samples, it’s first fed to backbone to update the original item ID embedding \( v_i \), then used to train CVAE and update \( w_\text{enc}, w_\text{dec}, w'_\text{enc} \). Updated \( v_i \) is also consumed in the training of CVAE at each step. Thus we claim that CVAE not only learns the pattern in historical data but also uses the information of update at each step to relieve the cold-start issue.

In inference phase of recent emerging items, their item ID embeddings \( v_i \) are not well trained. Therefore we obtain \( z' \) by sampling from \( N_i' \) which is generated by item side information and get enhanced warmed-up ID item \( \hat{v}_i' \) by passing \( z' \) to \( g_\text{dec} \) as Equation (7) and (8) shown. Then we could replace \( v_i \) with \( \hat{v}_i' \) for testing of recent emerging items. As marked in Figure 1, Equation (4) and (5) are operated in training of CVAE, while Equation (8) and (9) play a
Table 1: Model Comparison of cold-start effectiveness on two datasets (MovieLens-1M and Taobao Ad), under two backbones (DeepFM and Wide&Deep), three runs for each. The best improvements are highlighted in bold.

| Methods          | MovieLens-1M & Backbone: DeepFM | Dataset: Taobao Ad & Backbone: Wide&Deep |
|------------------|---------------------------------|------------------------------------------|
|                  | Cold phase                      | Warm-a phase                             |
|                  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  |
| DeepFM           | 0.7267 | 0.6231 | 0.7424 | 0.6383 | 0.7574 | 0.6593 | 0.7694 | 0.6608 |
| DropoutNet       | 0.7307 | 0.6319 | 0.7451 | 0.6443 | 0.7587 | 0.6511 | 0.7673 | 0.6590 |
| Meta-E           | 0.7297 | 0.6344 | 0.7444 | 0.6452 | 0.7544 | 0.6519 | 0.7633 | 0.6552 |
| MWUF             | 0.7316 | 0.6289 | 0.7462 | 0.6413 | 0.7589 | 0.6521 | 0.7701 | 0.6616 |
| CVAR (Init Only) | 0.7543 | 0.5581 | 0.7554 | 0.5561 | 0.7741 | 0.5581 | 0.7955 | 0.5582 |
| CVAR             | 0.7419 | 0.4735 | 0.7972 | 0.6789 | 0.9021 | 0.6856 | 0.9481 | 0.6876 |

| Methods          | MovieLens-1M & Backbone: Wide&Deep | Dataset: Movielens-1M & Backbone: Wide&Deep | Dataset: Movielens-1M & Backbone: DeepFM |
|------------------|---------------------------------|------------------------------------------|------------------------------------------|
|                  | Cold phase                      | Warm-a phase                             | Warm-b phase                             | Warm-c phase                             |
|                  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  | AUC  | F1  |
| DeepFM           | 0.6089 | 0.1360 | 0.6412 | 0.1427 | 0.6627 | 0.1455 | 0.6627 | 0.1464 |
| DropoutNet       | 0.6082 | 0.1378 | 0.6412 | 0.1424 | 0.6590 | 0.1477 | 0.6509 | 0.1456 |
| Meta-E           | 0.6082 | 0.1360 | 0.6412 | 0.1424 | 0.6220 | 0.1357 | 0.6220 | 0.1401 |
| MWUF             | 0.6082 | 0.1360 | 0.6412 | 0.1424 | 0.6220 | 0.1357 | 0.6220 | 0.1401 |
| CVAR (Init Only) | 0.6561 | 0.1515 | 0.6746 | 0.1628 | 0.7126 | 0.1555 | 0.7126 | 0.1555 |
| CVAR             | 0.6051 | 0.1368 | 0.6220 | 0.1457 | 0.6450 | 0.1495 | 0.6356 | 0.1511 |

3 EXPERIMENTS

3.1 Offline Experiments

3.1.1 Experiment Setup. In this section, we will introduce the offline experiment setup.

Public Datasets. For offline experiments, We evaluate CVAR on the following two public datasets MovieLens-1M\(^1\) and Taobao Ad\(^2\):

- MovieLens-1M: One of the most well-known recommendation benchmark datasets. The data consists of 1 million movie ranking instances over thousands of movies and users. Features of a movie include its title, year of release, and genres which are seen as item side information. Titles and genres for CTR prediction, with 4 categorical attributes as side information, including category ID, campaign ID, brand ID, and advertiser ID. Each user has features including the user’s ID, age, gender, and occupation.

- Taobao Display Ad Click: It randomly samples 1140000 users from 26 million ad display / click records on Taobao\(^3\) website to construct the dataset. We take each Ad as a item for CTR prediction, with 4 categorical attributes as side information, including category ID, campaign ID, brand ID, and advertiser ID. Each user has features including Micro group ID, cms_group_id, gender, age, consumption grade, shopping depth, occupation and city level.

Dataset Split. To demonstrate the recommendation performance in both cold-start and warm-up phases, we conduct the experiments by splitting the datasets following [26] and [57]. We divide items into two groups, old and new based on their frequency, where items with more than \(N\) labeled instances are old and others are new. We use \(N\) of 200 and 2000 for MovieLens-1M and Taobao Ad data. Note that the ratio of new items to old items is approximately 8:2, which is similar to the definition of long-tail items [3]. Besides, new item instances sorted by timestamp are divided into four groups denoted as warm-a, -b, -c and test set following [26] and [57].

Backbones and Baselines. Because CVAR is model-agnostic, it can be applied to various existing models in the Embedding & MLP paradigm. Thus we conduct experiments upon the following representative backbones: FM [30], DeepFM [12], Wide&Deep [4], DCN [42], IPNN [29], OPNN [29]. Meanwhile we choose some State-Of-The-Art (SOTA) methods for the item cold-start problem as baselines: DropoutNet [39], Meta embedding-Meta-E [26], MWUF [57]. We reproduce each baseline based on open source code or their publications if the code is unavailable. We open all of the related source code on Github\(^4\).

Implementation Details. For a fair comparison, we use the same setting for all methods. The MLPs in backbones and cold-start modules use the same structure with two dense layers (hidden units 16). The embedding size of each feature is fixed to 16. Learning rate and mini-batch size are set to 0.001 and 2048 respectively. At the inference stage, \(x_{freq}\) of new emerging items is set to the largest item frequency in the corresponding dataset to generate warmed-up embeddings. Training is done with Adam [15] optimizer over shuffled samples. In experiments, we firstly use old item instances to pretrain the backbone model as well as the cold-start module and evaluate on the test set(Initialization phase). Then we in turn feed warm-a, -b, -c data to train the backbone or CVAR and evaluate models on the test set step by step. We take the AUC score [21] and the F1 score [14] as evaluation metrics.

3.1.2 Experimental Results. We compare the cold-start effectiveness of CVAR with backbone and three SOTA cold-start baselines including DropoutNet, Meta-E, MWUF. Meanwhile, we choose two most famous CTR prediction methods as backbones, DeepFM and Wide&Deep. To evaluate the quality of initial embedding generated by CVAR and further prove the effectiveness of CVAR in leveraging the changing id embedding for better warm-up, we conduct a version of contrast experiment for CVAR denoted as CVAR (Init Only), where CVAR only plays a role in initialization phase and is disabled in following three warm-up phases. We conduct experiments on two datasets and evaluate the mean results over three runs.

---

\(^1\)http://www.grouplens.org/datasets/movielens

\(^2\)https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

\(^3\)https://www.taobao.com

\(^4\)https://github.com/BestActionNow/CVAR
The main experimental results are shown in Table 1. Notice that CVAR (Init Only) outperforms other baselines in most cases, which indicates a high quality of initial embeddings generated by CVAR. Moreover, superior performance of CVAR (Init Only) proves that distribution alignment in latent space is better than directly mapping side information to item id which is adopted in Meta-E and MWUF. Except for a better initialization, CVAR can significantly improve the prediction performance in warm-up stages which is proven by the outstanding results of CVAR in Table 1. This phenomenon demonstrates that CVAR can indeed produce high quality warmed-up embedding based on the evolving item ID embedding learned by backbone in warm-up stage.

Method compatibility. Since CVAR is model-agnostic, we conduct experiments in more scenarios to verify its compatibility. Results on six popular backbones and two datasets in Figure 2 demonstrate the compatibility and robustness of CVAR.

Comparison of different frequency condition As mentioned before, \( x_{freq} \) is set as the condition of decoder in CVAR and has a direct impact on the distribution of ID embedding. Considering the final \( x_{freq} \) is uncertain for new emerging items at inference stage, we compare the performance of CVAR with different \( x_{freq} \). Because \( x_{freq} \) is normalized before using, we conduct five experiments with \( x_{freq} \) equal 0.01, 0.1, 0.25, 0.5 and 1. Experimental results are shown in Table 2. It’s shown that performance of CVAR increases gradually with the increase of \( x_{freq} \), which explains why we set \( x_{freq} \) as a huge value at inference stage for cold start.

3.2 Online A/B tests

To verify the effectiveness of CVAR, we further conduct online A/B tests [16] for 7 days on Tencent News recommendation platform. As usually adopted in industry recommendation, the whole coarse-to-fine recommendation process can be divided into four stages: candidate generation, coarse-grained ranking, fine-grained ranking, and re-ranking. In our industrial scenario, CVAR is applied to the ranking stage whose backbone is MMOE [23, 37]. Consistent with the four phases in offline experiments, we group online items into four groups: cold, warm-a, -b, -c, with increasing exposed frequency. In our system, there are two mediums of news: Article and Video. We focus on the following metrics [11] (from high importance to low): Exposure Rate, Watch Time, Page(Video) Views. Exposure Rate measures the distribution percentage of the item group. Watch Time and Page(Video) Views reflect how users are attracted by recommended content.

We show the online results on various item groups in Table 3. It’s apparent that metrics of cold items are significantly improved, which proves the effectiveness of CVAR. As expected, the gain of CVAR gradually diminishes as the exposed frequency increases. Moreover, exposed items’ Gini coefficient [56] on various item categories reduces from 0.7413 to 0.7369 after applying CVAR, which indicates CVAR could alleviate the Matthew Effect [41] to some extent.

4 CONCLUSION

In this paper, we proposed the model-agnostic CVAR for item cold-start which uses latent variables to learn a distribution over side information and generates desirable ID embeddings using a conditional decoder. For better cold-start performance, CVAR not only learns the pattern in historical data but also leverages the continuously updating item ID embedding from the backbone to generate enhanced warmed-up embeddings with superior quality. From the aspect of industrial practice, we claim that additional strict data requirements of cold-start methods will make the deployment process rather difficult in online scenario. Thus CVAR is designed to be trained with the same raw samples as training main prediction model. Note that the proposed CVAR is a general framework that can be applied to various backbones. Finally, extensive offline experiments on public datasets and online A/B tests show the effectiveness and compatibility of CVAR.
