Alleviating Cold-start Problem in CTR Prediction with A Variational Embedding Learning Framework

Xiaoxiao Xu, Chen Yang, Qian Yu, Zhiwei Fang, Jiaxing Wang,
Chaosheng Fan, Yang He, Changping Peng, Zhangang Lin, Jingping Shao
Business Growth BU, JD.com
{xuxiaoxiao1,yangchen198,yuqian81,fangzhiwei2,wangjiaxing41}@jd.com
{fanchaosheng1,landy,pengchangping,linzhangang,shaojingping}@jd.com

ABSTRACT

We propose a general Variational Embedding Learning Framework (VELF) for alleviating the severe cold-start problem in CTR prediction. VELF addresses the cold start problem via alleviating over-fits caused by data-sparsity in two ways: learning probabilistic embedding, and incorporating trainable and regularized priors which utilize the rich side information of cold start users and advertisements (Ads). The two techniques are naturally integrated into a variational inference framework, forming an end-to-end training process. Abundant empirical tests on benchmark datasets well demonstrate the advantages of our proposed VELF. Besides, extended experiments confirmed that our parameterized and regularized priors provide more generalization capability than traditional fixed priors.

CCS CONCEPTS

• **Computational advertising**; • **Recommender systems**;

KEYWORDS

CTR prediction, Cold-start, Embedding learning, Variational inference

ACM Reference Format:
Xiaoxiao Xu, Chen Yang, Qian Yu, Zhiwei Fang, Jiaxing Wang,, Chaosheng Fan, Yang He, Changping Peng, Zhangang Lin, Jingping Shao. 2022. Alleviating Cold-start Problem in CTR Prediction with A Variational Embedding Learning Framework. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

In spite of the impressive development of deep learning in recent decades, cold-start problem keeps becoming a stubborn obstacle in many tasks. Typically, the data scarcity is the major cause of the cold-start problem, and the help by refining model structure is limited. As exemplified by Figure 1, the severe cold-start problem in online advertising is usually caused by two issues: 1) the drastic long-tail phenomenon which is a widely recognized fact and 2) the remarkable real-time new user and Ad updating.

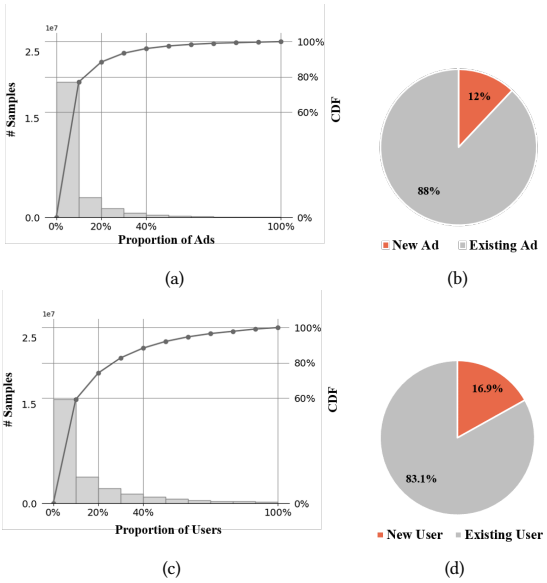

Figure 1: Statistics are from Taobao Display Ad Click [1]dataset: The severe long-tail distributions of both (a) Ads and (c) users are shown, and there are above (b) 12% new Ads and (d) 16.9% new users in the daily updates.

Being the most successful application of deep neural networks in online advertising, Click-through Rate (CTR) prediction task is also troubled by cold-start problems.

The state-of-the-art deep CTR models mostly adopt an Embedding&Network paradigm as illustrated in Figure 2, in which Embedding module works as a representative mapper [6, 8, 17]. Embedding module maps each distinct feature value to a low-dimensional dense embedding vector, where discrete features cover all the categorical features and discretized numerical features. The number of trainable parameters in deep CTR models is heavily concentrated in Embedding module, and the Embedding module determines the input distribution of the subsequent feature interaction module and MLP module. It is considerably data demanding to train a good Embedding module, and this makes it a challenging task in recommendation systems to provide reasonable embeddings for users and Ads with few or no support samples [20]. Therefore, enhancing the generalization ability and robustness of the Embedding Module is critical for alleviating the severe cold-start problem in CTR prediction.

[1]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56

WWW '22, April 25–29, 2022, Lyon, France.

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

Obtaining reasonable embedding for cold-start user and Ad is a challenging task and has been an active research area [4, 12, 18–20, 22, 24–26, 28, 33]. The existing work mostly concentrates on two kinds of methods, i.e., content-based methods [18, 22, 24–26, 28] and meta-learning involved methods [4, 12, 19, 20, 33]. Content-based methods introduce richer attributes of user or Ad to obtain a more robust embedding for user or Ad. Meta-learning is involved to transfer knowledge from other users or Ads to the cold-start ones by carefully designed training procedure and sample partition. Both content-based and meta-learning involved methods have been confirmed to be effective. However, these methods are all based on point estimate, i.e., trying to locate a reliable single point in the embedding space for each user and Ad. Previous research has shown that point estimate has a huge risk to result in isolated and unreliable embedding for cold-start user and Ad because of the scarce training samples [30]. In addition, the model for embedding point estimation is prone to overfit unless carefully designed regularization is equipped for parameter tuning [23].

To make better use of the limited data to obtain more reliable embedding for cold-start users and Ads and avoid overfitting, we propose a general Variational Embedding Learning Framework (VELF). VELF regards the embedding learning as distribution estimate instead of point estimate. Through building a probabilistic embedding framework based on Bayesian inference, the statistical strength among users and Ads can be shared, especially by the cold-start ones. Bayesian approach has been approved to be more robust regarding to data scarcity [15] and more interpretable. The distributions are estimated via variational inference (VI) which can avoid the computational intractability. To better share the global and statistical strength among users and Ads, we propose to parameterize priors with neural network and the attributes of users and Ads as inputs. A regularized and parameterized prior framework combining the parameterized prior with the fixed standard normal prior is proposed to further avoid overfitting. VI solves distribution estimate as an optimization problem, thus the parameters of the probabilistic embedding framework and the following discriminative CTR prediction network are jointly learned enjoying an end-to-end manner.

In this paper we focus on CTR prediction in the scenario of display advertising. Methods discussed here can be applied in similar scenarios suffering cold-start problems, such as personalized recommendation, sponsored search, etc. The major contributions in this paper are:

- We propose a general Variational Embedding Learning Framework (VELF) with an interpretable probabilistic embedding generation process to alleviate the cold-start problem in CTR prediciton. The embedding distributions and the discriminative CTR predicition network parameters are learned end-to-end.
- Novel parameterized and regularized priors naturally utilizing the rich side information are designed to further improve the generalization ability of our model.
- Extensive experiments are conducted on three benchmark datasets. Results verify the effectiveness of our proposed VELF and the superiorities of proposed parameterized and regularized priors.

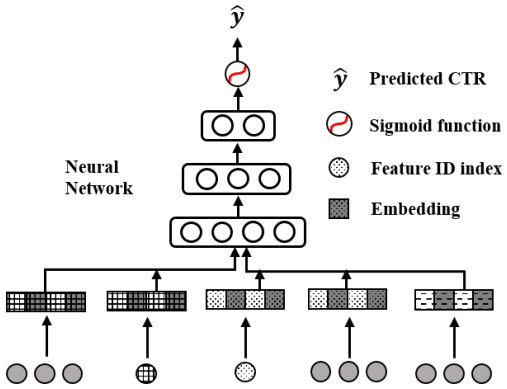

Figure 2: An illustration of the traditional Embedding&Network paradigm structure for CTR prediction.

2 RELATED WORK

In this section, we will introduce the related work from two aspects: Cold-start Recommendation and Variational Inference in Recommendation.

Cold-start Recommendation: Cold-start problem is usually caused by two issues: the prevailing long-tail phenomenon and the continuous update of new users and Ads. It is challenging to make recommendations for cold-start users and Ads because of the data limitation. In this paper, we focus on the cold-start problem alleviation in CTR prediction task by embedding optimization.

To alleviate cold-start problem in CTR prediction by improving embedding generalizaiton ability, Content-based methods and meta-learning involved methods fall into the same domain with our work. Content-based methods introduce side information for cold-start IDs, i.e., using user and item attributes[18, 22, 24–26, 28], and relational data[16, 31]. DropoutNet[28] is the representative of the content-based methods. They can improve the cold-start performance comparing to classic methods that do not use those rich features. However, they do not improve user ID and Ad ID directly. Meta-learning is intent to learn the general knowledge across similar learning tasks, so as to rapidly adapt to new tasks based on a few examples. Meta-learning methods have been widely proposed for cold-start recommendation, e.g., learning a meta-learner to better initialize CTR prediciton models [4, 20], utilizing Meta-Embedding [12, 33], in which MWUF [33] is the State-of-the-art. These methods have been validated to be effective, but they need carefully tuning with carefully designed training procedures.

Above all, content-based methods and meta-learning involved methods are all based on point estimate which still has a huge risk to result in isolated and unreliable embedding for cold-start user and Ad [30]. In addition, the model for embedding point estimate is prone to overfit [23].

Variational Inference in Recommendation: Variational inference (VI) has been applied in recommendation but coupling with AutoEncoders, i.e., Variaitonal AutoEncoders (VAEs) [10]. In recommendation, VAEs concentrate on Collaborative filtering and are trying to model the uncertainty of users and items representations, and then collectively reconstructing and predicting user preferences [1, 9, 15, 27]. Different from these methods, we apply VI to discriminative models, i.e., CTR prediction task, to alleviate

u, z^u	user id and its embedding
i, z^i	Ad id and its embedding
$c(\cdot)$	the number of attributes of ID
$g_{\boldsymbol{\phi}}$	parameterized function of embedding module
$f_{\boldsymbol{\theta}}$	parameterized function of MLP module
$q_{\boldsymbol{\phi}_q}(\boldsymbol{z}\|\boldsymbol{x})$	approximated posterior distributation
$p_{\boldsymbol{\phi}_p}$	prior distributions of embedding distribution

Table 1: Important notations.

cold-start problem. In our work, VI is the technique we choose to avoid the computationally intractable problem in distribution estimate for users and items embedding by Bayesian inference.

3 METHOD

In Section 3.1, we first review the background of CTR prediction, the fundamentals of Variational Inference and the cold-start issues of Point Estimate. Then we describe Distribution Estimate in our proposed variational embedding learning framework in Section 3.2. We further dig into the details of the implementations during training in Section 3.3 and predicting in Section 3.4. The notations are summarized in Table 1.

3.1 Preliminaries

3.1.1 CTR Prediction Problem Formulation. Given an user, a candidate Ad and the contexts in an impression scenario, CTR prediction, is to infer the probability of a click event. The CTR prediction model is mostly formulated as a supervised logistic regression task and trained with an i.i.d. dataset $\mathcal{D}$ collected from historic impressions. Each instance $(\boldsymbol{x}, y) \in \mathcal{D}$ contains the features $\boldsymbol{x}$ implying the information of $\{user, Ad, contexts\}$, and the label $y \in \{0, 1\}$ observed from user implicit feedback. Let u denotes the user ID index, i denotes the Ad ID index, $c(u)$ and $c(i)$ represents the course features, i.e., attributes of u and i, the instance features $\boldsymbol{x}$ can be expressed as:

$$\boldsymbol{x} = [u, c(u), i, c(i), contexts] \tag{1}$$

where contexts contain the scene information such as the positions, time, etc.

With the measurable progress of the research and application of neural network, most recent CTR prediction models share an Embedding and Multi-Layer Perceptron (MLP) paradigm. Specifically in each instance, $\boldsymbol{x} \in \mathbb{N}^m$ is the vector of feature ID indexes, and m means the total number of selected features. Because the selected features are ID indexes, they have to be encoded into real-value to apply optimization methods. Embedding Module solves this problem by mapping these ID indexes into low-dimensional representations and concatenating them to form the input of MLP Module afterwards, i.e.,

$$\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{x}) \tag{2}$$

where $g_{\boldsymbol{\phi}}(\cdot)$ refers to the function of the Embedding Module, and let $\boldsymbol{\phi}$ denotes its parameters. Subsequently, the estimated CTR $\hat{y}$ can be obtained by the following discriminative model,

$$\hat{y} = \sigma(f_{\boldsymbol{\theta}}(\boldsymbol{z})) \tag{3}$$

where $f_{\boldsymbol{\theta}}(\cdot)$ refers to the function of the MLP Module which is parameterized by $\boldsymbol{\theta}$, and $\sigma(\cdot)$ is the sigmoid activation function. The model parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are learned by maximizing the objective function $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$ with gradient-based optimization methods. In the traditional point estimate CTR prediction model, the objective function is equal to the negative log-likelihood $l(\boldsymbol{\phi}, \boldsymbol{\theta})$:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= l(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ &\equiv -y\log\hat{y} - (1-y)\log(1-\hat{y})\end{aligned} \tag{4}$$

3.1.2 Variational Inference. Variational inference is an analytical approximation technique to learn posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$ of latent variable $\boldsymbol{z}$ conditional on the observed variable $\boldsymbol{x}$. Variational inference can work well with deep learning by formulating the Bayesian inference problem in deep learning as an optimization-based approach. Thus, the stochastic gradient descent optimization methods can be adopted.

Now we summarize the fundamentals of variational inference. It is clear the posterior can be formulated as $p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{z})/p(\boldsymbol{x})$ based on Bayes rule. However, the marginal likelihood $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$ has no analytic solution or efficient estimator. To avoid the computational intractability, variational inference obtain the best approximate posterior distributation $q_{\boldsymbol{\phi}_q}(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z}|\boldsymbol{x})$ by maximizing the Evidence Lower Bound(ELBO) with respect to the variational parameters $\boldsymbol{\phi}_q$:

$$ELBO(\boldsymbol{\phi}_q) = \mathbb{E}(\log p(\boldsymbol{x}|\boldsymbol{z})) - D_{KL}(q_{\boldsymbol{\phi}_q}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) \tag{5}$$

3.1.3 Cold-start Issues of Point Estimate. In prior point estimate methods, Embedding module explicitly maps user ID and Ad ID into a low-dimension embedding space where the similar IDs are expected to be close. However, the embedding points for cold-start users and Ads tend to be isolated because of the ubiquitous data sparsity problem. To alleviate the probelm, in addition to the only similarity objective supervision introduced by collaborative filtering mechanism with interactions, the existing achievements are enlighten to use content-based and meta-learning based methods with IDs' attributes.

However, there are two important issues: 1) The attributes of users and Ads are only exploited to be as the ID point initialization before training or the fixed final representation of the ID point for inference. Thus the embedding of cold-start user or Ad is still risky to be isolated during training. 2) The point estimate suffers the overfitting problem. To further alleviate those problems, we are motivated to concentrate on estimating distributions for each user ID u and Ad ID i, which exploits the global knowledge during end-to-end training and is much more interpretable. Also, it is confirmed in our work that our proposed distribution estimate method is more effective and robust than point estimate methods when data is limited.

3.2 Distribution Estimate

In this section, we focus on addressing the theoretical essentials of our proposed methods, the implementation details will be covered in Section 3.3 and 3.4.

3.2.1 Variational Embedding Framework in CTR Prediction. We propose Variaitonal Embedding Learning Framework (VELF) aiming to predict the distribution of each user and Ad embedding. In this way, the model to be learned is regarded as $p_{\boldsymbol{\phi},\boldsymbol{\theta}}(y|\boldsymbol{x}, \boldsymbol{z})$, and $\boldsymbol{z}$ denotes the unobserved latent variables, i.e., embedding space, and

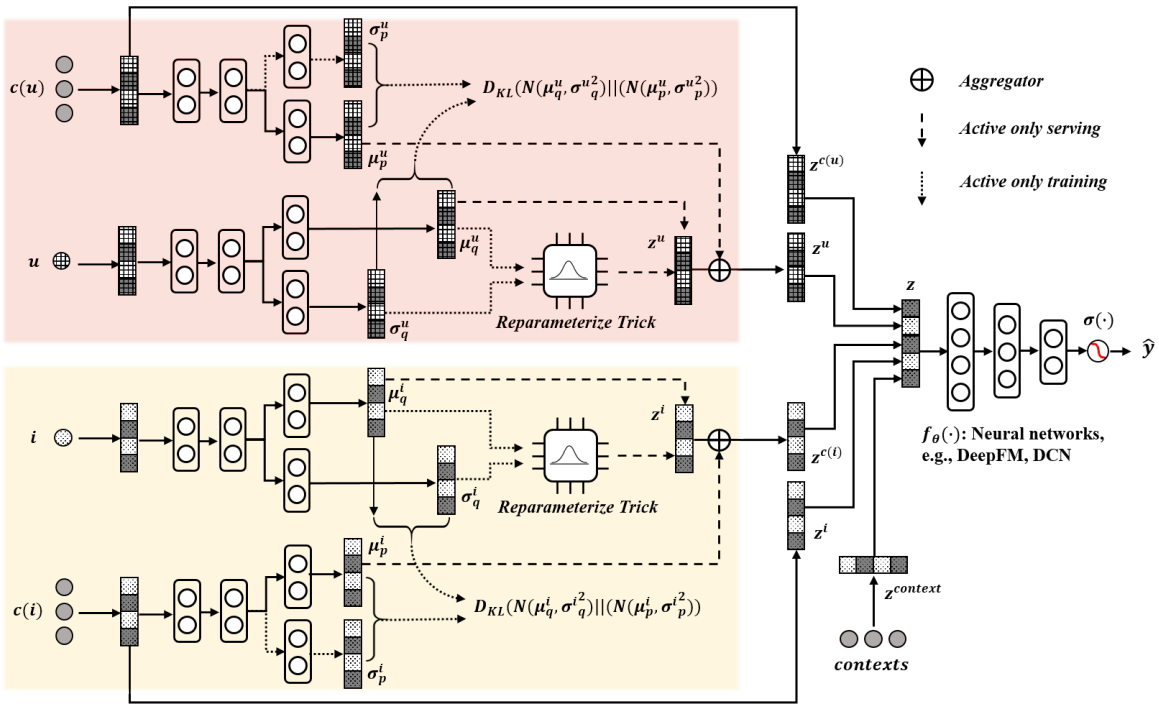

Figure 3: An illustration of the variational embedding learning framework for CTR prediction.

the distribution that has to be estimated is the posterior of $\boldsymbol{z}$ which is termed as $p(\boldsymbol{z}|\boldsymbol{x})$. We adopt Gaussian assumptions for all the distributions in our VELF.

Variational inference (VI) is chosen to obtain the approximate posterior distributation $q_{\boldsymbol{\phi}_q}(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z}|\boldsymbol{x})$, because it is efficient to be parameterized and computed by neural network. According to Section 3.1.2, it is clear that VI casts distribution estimate for latent variables $\boldsymbol{z}$ as an optimization problem. Parameterizing the probabilistic models in VI by neural networks, the computationally scalable stochastic gradient-based optimizing methods can be applied [11].

In VELF, our objective function is naturally equal to the Evidence Lower Bound (ELBO):

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}_q, \boldsymbol{\theta}) &= ELBO(\boldsymbol{\phi}_q, \boldsymbol{\theta}) \\ &\equiv \mathbb{E}(\log p(\boldsymbol{x}|\boldsymbol{z})) - D_{KL}(q_{\boldsymbol{\phi}_q}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))\end{aligned} \tag{6}$$

According to Equation 6, our optimization goal includes two terms. The first term tries to maximize the likehood to improve the confidence of prediction, and the second term tries to find the approximate posterior distribution by minimizing the KL divergence. For binary dataset $\mathcal{D}$ in our CTR prediction scenario, the confidence of prediction $\mathbb{E}(\log p(\boldsymbol{x}|\boldsymbol{z}))$ is calculated as Log-loss $l(\boldsymbol{\phi}, \boldsymbol{\theta})$ [10]. Adopting the same perspective as [15] that the KL divergence term can be viewed as regularization, we introduce a parameter α to control the trade-off between how well the model can fit the data and how close the approximate posterior is to the prior $p(\boldsymbol{z})$ during training. To reduce the time-consuming for selecting α, we also adopt the annealing method similar to [15]: we start with α=0, and gradually increase α to 1.

Now, let us cover the details of $p(\boldsymbol{z})$ which is another crux of our method. $p(\boldsymbol{z})$ represents the prior distribution of the latent embedding $\boldsymbol{z}$ in Bayes Learning, which is mostly assigned to a certain normal Gaussian distribution. We argue that fixed prior limits the generalization capability of our method due to the huge discrepancy among dissimilar users and Ads. In our proposed method, we parameterize $p(\boldsymbol{z})$ as $p_{\boldsymbol{\phi}_p}(\boldsymbol{z}|\boldsymbol{c})$ by neural networks with the coarse features $\boldsymbol{c}$ of user ID and Ad ID as inputs, where $\boldsymbol{\phi}_p$ denotes the specified neural networks parameters.

In this way, we can make full use of the information in dataset to obtain reasonable priors. IDs with the similar attributes can naturally cluster together in the latent embedding space, because they are sampled from similar distributions which are restricted to stay close to the similar prior distributions by the KL divergence regularization. Thus, the global knowledge in each cluster can be shared by cold-start IDs with few samples that it contains. With the global knowledge, even the cold-start IDs can obtain reasonable embeddings.

Finally, our maximization goal can be rewritten as:

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = l(\boldsymbol{\phi}, \boldsymbol{\theta}) - \alpha \cdot D_{KL}(q_{\boldsymbol{\phi}_q}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\phi}_p}(\boldsymbol{z})) \tag{7}$$

where $\boldsymbol{\phi} = [\boldsymbol{\phi}_q, \boldsymbol{\phi}_p]$.

3.2.2 Mean-field Variational Embedding Framework. In this paper, we aim to alleviate the cold-start problem both for users and Ads . There are two different latent variables, i.e., user latent embedding $\boldsymbol{z}^u$ and Ad latent embedding $\boldsymbol{z}^i$. On the strength of Mean-field Theory [2], we suppose $\boldsymbol{z}^u$ and $\boldsymbol{z}^i$ are mutually independent and each governed by distinct factors in the variational density. Then

our maximization goal turns into:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= l(\boldsymbol{\phi}, \boldsymbol{\theta}) \\ &-\alpha \cdot (D_{KL}(q_{\boldsymbol{\phi}_q^u}(\boldsymbol{z}^u|u)||p_{\boldsymbol{\phi}_p^u}(\boldsymbol{z}^u)) \\ &+ D_{KL}(q_{\boldsymbol{\phi}_q^i}(\boldsymbol{z}^i|i)||p_{\boldsymbol{\phi}_p^i}(\boldsymbol{z}^i)))\end{aligned} \tag{8}$$

where $\boldsymbol{\phi} = [\boldsymbol{\phi}_q^u, \boldsymbol{\phi}_q^i, \boldsymbol{\phi}_p^u, \boldsymbol{\phi}_p^i]$.

3.2.3 Regularized Priors. As mentioned before, we introduce unfixed parameterized priors for ID with ID's features as inputs. Thus we can make full use of the information in dataset to obtain reasonable priors and facilitate the knowledge sharing among IDs with similar attributes. However, the parameterized prior technique still has a risk of overfitting by introducing additional parameters of distributions. To mitigate the overfitting risk, we propose to regularize priors by forcing the parameterized priors to be close to a standard normal hyper-prior:

$$\begin{aligned}p(\boldsymbol{z}^u) &= \mathcal{N}(0, \boldsymbol{I}^u) \\ p(\boldsymbol{z}^i) &= \mathcal{N}(0, \boldsymbol{I}^i)\end{aligned} \tag{9}$$

With an additional component to the KL-divergence term, We can rewrite our objective function as:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}^u, \boldsymbol{\phi}^i, \boldsymbol{\theta}) &= l(\boldsymbol{\phi}^u, \boldsymbol{\phi}^i, \boldsymbol{\theta}) \\ &-\alpha \cdot (D_{KL}(q_{\boldsymbol{\phi}_q^u}(\boldsymbol{z}^u|\boldsymbol{x}^u)||p_{\boldsymbol{\phi}_p^u}(\boldsymbol{z}^u)) + \\ &\quad D_{KL}(q_{\boldsymbol{\phi}_q^i}(\boldsymbol{z}^i|\boldsymbol{x}^i)||p_{\boldsymbol{\phi}_p^i}(\boldsymbol{z}^i))) \\ &-\alpha \cdot (D_{KL}(p_{\boldsymbol{\phi}_p^u}(\boldsymbol{z}^u)||p(\boldsymbol{z}^u)) + \\ &\quad D_{KL}(p_{\boldsymbol{\phi}_p^i}(\boldsymbol{z}^i)||p(\boldsymbol{z}^i)))\end{aligned} \tag{10}$$

3.3 Training with Distribution

In this chapter, we describe the training implementation details of VELF, which is illustrated in Figure 3.

Here, we take the obtaining of user embedding $\boldsymbol{z}^u$ as an illustration, and Ad embedding $\boldsymbol{z}^i$ can be obtained in the same way. With parameterization by neural network, the posterior distributions and prior distributions can be obtained by data-dependent functions:

$$q_{\boldsymbol{\phi}_q^u}(\boldsymbol{z}^u|u) = \mathcal{N}(\mu_q^u(u), \sigma_q^{u\,2}(u)) \tag{11}$$

$$p_{\boldsymbol{\phi}_p^u}(\boldsymbol{z}^u) = p_{\boldsymbol{\phi}_p^u}(\boldsymbol{z}^u|c(u)) = \mathcal{N}(\mu_p^u(c(u)), \sigma_p^{u\,2}(c(u))) \tag{12}$$

As shown in Figure 3, the posterior distribution parameters μ_q^u and σ_q^u are computed from feature ID u with DNNs, and prior distribution parameters μ_p^u and σ_p^u are computed from the attributes of u with DNNs.

In VELF, the latent embedding of user ID $\boldsymbol{z}^u$ are sampled from the estimated posteriors via reparameterization trick. Given an instance $(\boldsymbol{x}, y)$, the resulting user embedding for each sampling can be calculated as:

$$\begin{aligned}\boldsymbol{z}^u &= \mu_q(u) + \sigma_q(u) \odot \boldsymbol{\epsilon}^u \\ \boldsymbol{\epsilon}^u &\sim \mathcal{N}(0, \boldsymbol{I})\end{aligned} \tag{13}$$

As illustrated in Figure 3, the embedding of user, Ad, context, and attributes of user and Ad, are concatenated to obtain the input embedding z for the discriminative model:

$$\hat{y} = \sigma(f_{\boldsymbol{\theta}}(\text{concat}(\boldsymbol{z}^u, \boldsymbol{z}^i, \boldsymbol{z}^{c(u)}, \boldsymbol{z}^{c(i)}, \boldsymbol{z}^{context}))) \tag{14}$$

The resulting Log-loss for sample $(\boldsymbol{x}, y)$ can be obtained by:

$$l(\boldsymbol{\phi}, \boldsymbol{\theta}) = \frac{1}{L}\sum_{k=1}^{L}(-y\log\hat{y}_{(k)} - (1-y)\log(1-\hat{y}_{(k)})) \tag{15}$$

where each $\hat{y}_{(k)}$ is calculated by Equation 14 with randomly sampled $\boldsymbol{\epsilon}^u$ and $\boldsymbol{\epsilon}^i$, and L is the total Monte Carlo sampling number for each instance and is fixed to be 1 in this paper. The KL-divergence terms in *ELBO* can be computed and differentiated without estimation following the given definition for Gaussian distributions:

$$D_{KL}(q||p) = \log\frac{\sigma_p}{\sigma_q} + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} \tag{16}$$

We apply Equation 15 and Equation 16 to Equation 10, yielding the differentiable *ELBO*, i.e., our objective function. By maximizing the differentiable *ELBO*, the variational parameters $\boldsymbol{\phi}$ and the discriminative model parameters $\boldsymbol{\theta}$ are jointly learned in an end-to-end manner. The computationally scalable mini-batch stochastic gradient descent methods are adopted during the training procedure.

3.4 Predicting with Distribution

As shown in Figure 3, given a VELF model trained following Section 3.3, we make predictions using the centers of the estimated posteriors and the parameterized priors, i.e., the means of distributions. Here, we take the obtaining of user embedding $\boldsymbol{z}^u$ as an illustration, and Ad embedding $\boldsymbol{z}^i$ can be obtained in the same way. The means of the parameterized priors are used to make up for the unreliable posteriors of the awfully infrequent or new IDs.

$$\boldsymbol{z}^u = g(u)\mu_q(u) + (1 - g(u))\mu_p(c(u)) \tag{17}$$

As a variant of sigmoid function with the statistics of u as inputs, $g(u)$ is designed to control the weights for μ_q^u and $\mu_p(u)$ to form the final representation of u during inference:

$$g(u) = \frac{1}{1 + e^{-\mathcal{F}(u)+\epsilon}} \tag{18}$$

where $\mathcal{F}(u)$ is the accumulated frequency of u in the training dataset and ϵ is a small constant for numerical stability. In this way, a certain amount of flexibility is built. The convincing estimated posterior leads the role for a frequent ID, while the convincing prior balances the unreliable posterior for a new or awfully infrequent ID. As shown in Figure 3, $\hat{y}$ is then calculated by Equation 14.

It is easy to see the advantage of our VELF model. We can effectively make predictions for infrequent and new users and Ads by evaluating posterior and prior distributions end to end. For an infrequent user or Ad, the global knowledge among the similar and frequent users or Ads can be shared during training and inference. The center of the global knowledge is used to represent a new user or Ad to improve the accuracy when inference.

4 EXPERIMENTS

In this section, we conduct experiments with the aim of answering the following three research questions:

RQ1 How does VELF perform compared to the existing cold-start methods from the perspective of embedding?

RQ2 How does VELF perform when plugged into various network backbones?

RQ3 What are the effects of distribution estimate, parameterized and regularized priors in VELF?

4.1 Dataset

We evaluate the performance of our proposed approaches on three publicly available datasets:

- **MovieLens-1M**[2]**:** One of the most well-known benchmark dataset. The dataset is made up of 1 million movie ranking instances over thousands of movies and users. The movie ratings are transferred into binary (The ratings at least 4 are turned into 1 and the others are turned into 0).
- **Taobao Display Ad Click**[3]**:** The dataset consists of 26 million ad display / click records generated by 1.14 million users on Taobao website within 8 days.
- **CIKM2019 EComm AI**[4]**:** An E-commerce recommendation dataset contains 62 million instances, each of which consists of an item, a user, and a behavior label ('pv,' 'buy,' 'cart,' 'fav'). To match the issue, we convert the instance label to binary (1/0 indicates whether a user has purchased an item or not).

We now describe the training sets and test sets preparation.

Training sets: We follow the training setting commonly used in previous research works. For **MovieLens-1M**, we use the first 80% instances of each user ordered by time as training set [13] and further move the users with less than 30 reviews to test set. For **Taobao Display Ad Click**, the click data that generated in the first 7 days are used as training set, and the data in the last 1 day is the test set in our experiment [32]. For **CIKM2019 EComm AI**, we use the default training set of this dataset [33].

Test sets: For testing, we prepared 5 different test sets to assess recommendation performance for both new users/items and infrequent users/items. The corresponding definitions for each test set are given in Table 2. Note that the infrequent users/items proportion of the overall users/items is approximately 20%, which is similar to the definition of the long-tail [20].

The statistics of the three datasets can be found in Table 3, and the utilized features are listed in Table 2. In Table 2 and Table 3, 'Infreq' is short for 'Infrequent' and 'fea' is short for 'features'. The datasets we adopt contain one dataset of online advertising and two datasets of personalized recommendation. Thus we generally call 'Ad' as well as 'Item' as 'Item'.

4.2 Baselines

We divide our baselines into two groups based on their methods.

The first group contains state-of-the-art approaches for dealing with the cold-start problem. (1) DropoutNet [28] is a famous cold-start approach that uses the average representations of interacting items/users to improve user/item representations. (2) MWUF [34] introduce Meta Scaling and Shifting Networks to construct scaling and shifting functions for each item, with the scaling function directly transforming cold item ID embeddings into warm feature space and the shifting function producing stable embeddings from noisy embeddings.

[2]http://www.grouplens.org/datasets/movielens/
[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=56
[4]https://tianchi.aliyun.com/competition/entrance/231721/introduction?lang=en-us

MovieLens-1M	Item fea	*title, year of release, genres*
	User fea	*unique ID, age, gender, occupation*
	New user	Users who commented less than 30 times.
	New item	Movies released after 1997.
	Infreq user	80% users ordered by the number of posed comment
	Infreq item	80% movies ordered by the number of users interacted with it.
Taobao Display Ad Click	Item fea	*Ad ID, category ID, campaign ID, brand ID, Advertiser ID, price*
	User fea	*user ID, Micro group ID, cms_group_id, gender, age, consumption grade, shopping depth, occupation, city level*
	New user	Users who exist only in the default test set.
	New item	Items that exist only in the default test set.
	Infreq user	60% users ordered by the number of the number of the related click
	Infreq item	80% items ordered by the number of users interacted with it.
CIKM2019	Item fea	*item ID, category ID, shop ID, brand*
	User fea	*user ID, gender, age, purchasing power*
	New user	Users who exist only in the default test set.
	New item	Items that exist only in the default test set.
	Infreq user	20% users ordered by the number of the number of the related click.
	Infreq item	80% items ordered by the number of users interacted with it.

Table 2: Features and Test set construction.

Dataset	MovieLens	Taobao Ad	CIKM2019
#user	6,040	1,141,729	1,050,000
#item	3,706	864,811	3,934,201
#training sample	630,602	21,929,927	58,751,493
#test (All)	369,607	3,099,508	3,677,047
#test (New user)	18,169	275,723	3,677,047
#test (New item)	196,059	87,894	114,906
#test (Infreq user)	177,380	391,007	81,964
#test (Infreq item)	137,508	369,561	570,590

Table 3: Statistics of datasets. #test stands for the number of instance in different test sets.

The common Feature-Crossing techniques developed for overall recommendation are included in the second group. The second group serves as baselines without cold-start alleviating component as well as backbones to test the generalization and adaptability of our proposed VELF. (1) DeepFM [7] is a deep recommendation method that learns both low- and high-level interactions between fields. (2) Wide&Deep [3] develop wide linear models and deep neural networks together to enhance their respective abilities. (3) DCN [29], which is based on DNN, explicitly applies feature crossing at each layer, eliminating the need for human feature engineering. (4) xDeepFM [14] generates feature interactions directly at the vector-wise level, allowing it to learn specific bounded-degree feature interactions explicitly at the low- and high-order levels. (5) PNN [21] employs a feature extractor to investigate feature interactions among inter-field categories.

4.3 Experimental Settings

4.3.1 Implementation Details. We utilize the same model settings for all approaches on each dataset to provide a fair comparison. For all the three datasets, we fix embedding size as 8 and DNN as 3 FC

Table 4: Model comparison on three datasets. We record the mean results over 5 runs. Std ≈ 0.1%, extremely statistically significant under unpaired t-test. * indicates the improvement is statistically significant at the significance level of 0.05 over the best baseline on AUC. 'Infreq' is short for 'Infrequent'.

	Methods	New user		New item		Infreq user		Infreq item		All	
		AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
MovieLens-1M	Wide & Deep	0.6771	5.0%	0.6488	16.8%	0.6955	4.7%	0.6786	4.9%	0.7276	3.0%
	PNN	0.6701	0.9%	0.6470	15.4%	0.6955	4.7%	0.6840	8.1%	0.7275	2.9%
	DCN	0.6785	5.9%	0.6460	14.6%	0.6946	4.2%	0.6778	4.5%	0.7280	3.2%
	xDeepFM	0.6781	5.6%	0.6476	15.9%	0.6958	4.8%	0.6799	5.7%	0.7294	3.8%
	DeepFM	0.6686	0.0%	0.6274	0.0%	0.6868	0.0%	0.6702	0.0%	0.7210	0.0%
	DropoutNet(DeepFM)	0.6640	-2.7%	0.6298	1.9%	0.6875	0.4%	0.6711	0.6%	0.7216	0.3%
	MWUF(DeepFM)	0.6712	1.5%	0.6573	23.5%	0.6991	6.6%	0.6886	10.8%	0.7342	6.0%
	VELF(DeepFM)	**0.7112***	**25.3%**	**0.7106***	**65.3%**	**0.7117***	**13.3%**	**0.7009***	**18.0%**	**0.7551***	**15.4%**
	Methods	New user		New item		Infreq user		Infreq item		All	
		AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
Taobao Display Ad Click	Wide & Deep	0.5573	-30.1%	0.5995	-8.3%	0.5713	-18.2%	0.5964	-12.5%	0.6204	-8.1%
	PNN	0.5535	-34.8%	0.6064	-1.9%	0.5547	-37.3%	0.5991	-10.1%	0.6140	-13.0%
	DCN	0.5843	2.8%	0.6141	5.2%	0.5873	0.1%	0.6157	5.0%	0.6256	-4.1%
	xDeepFM	0.5831	1.3%	0.6104	1.8%	0.5874	0.2%	0.6129	2.5%	0.6328	1.4%
	DeepFM	0.5820	0.0%	0.6085	0.0%	0.5872	0.0%	0.6102	0.0%	0.6310	0.0%
	DropoutNet(DeepFM)	0.5848	3.4%	0.6179	8.7%	0.5884	1.4%	0.6303	18.2%	0.6340	2.3%
	MWUF(DeepFM)	0.5819	-0.1%	0.6117	2.9%	0.5896	2.8%	0.6244	12.9%	0.6322	0.9%
	VELF(DeepFM)	**0.5895***	**9.1%**	**0.6220***	**12.4%**	**0.5998***	**14.4%**	**0.6332***	**20.9%**	**0.6394***	**6.4%**
	Methods	New user		New item		Infreq user		Infreq item		All	
		AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr	AUC	RelaImpr
CIKM2019 Ecomm AI	Wide & Deep	0.7467	0.0%	0.6877	0.1%	0.7451	0.3%	0.7139	0.3%	0.7467	0.0%
	PNN	0.7468	0.0%	0.6882	0.3%	0.7433	-0.4%	0.7145	0.6%	0.7468	0.0%
	DCN	0.7468	0.0%	0.6883	0.4%	0.7449	0.2%	0.7143	0.5%	0.7468	0.0%
	xDeepFM	0.7464	-0.1%	0.6867	-0.5%	0.7438	-0.2%	0.7132	0.0%	0.7464	-0.1%
	DeepFM	0.7467	0.0%	0.6876	0.0%	0.7443	0.0%	0.7132	0.0%	0.7467	0.0%
	DropoutNet(DeepFM)	0.7467	0.0%	0.6886	0.5%	0.7455	0.5%	0.7138	0.3%	0.7467	0.0%
	MWUF(DeepFM)	0.7483	0.6%	0.6887	0.6%	0.7450	0.3%	0.7164	1.5%	0.7483	0.6%
	VELF(DeepFM)	**0.7497**	**1.2%**	**0.6967***	**4.9%**	**0.7492**	**2.0%**	**0.7228***	**4.5%**	**0.7497**	**1.2%**

layers with 200 hidden units. Furthermore, for xDeepFM and DCN, we set the number of cross layer to 2. We optimize all approaches using mini-batch Adam, where the learning rate is searched from {1e-5, 5e-4, 1e-4, ..., 1e-2}. Furthermore, the batch size of all models is set to 256 for the MovieLens-1M dataset and 4096 for others.

4.3.2 Evaluation Metrics. AUC [5] is a common metric for both recommendation [34] and advertising [32]. It measures the goodness of order by ranking all the items with prediction. Thus following the cold-start work [20, 34], we adopt AUC as the main metric. In addition, we follow [32, 34] to introduce RelaImpr metric to measure relative improvement over models. For a random guesser, the value of AUC is 0.5. Hence, RelaImpr is defined as:

$$\text{RelaImpr} = \left(\frac{\text{AUC(measured model)} - 0.5}{\text{AUC(base model)} - 0.5} - 1\right) \times 100\% \quad (19)$$

4.4 Comparison with State-of-the-arts (RQ1)

We compare our VELF with the SOTA methods to alleviate cold-start problem from the perspective of embedding learning, i.e., DropoutNet [28] and MWUF [34]. Comparisons with state-of-the-arts are conducted on DeepFM [7] which is one of the most popular model structures used in industry. For more detailed and directed analysis, we also report the results of the second group baseline models mentioned before. Evaluations are conducted on three benchmark datasets to report the mean results over five runs. The results are shown in Table 4.

The effectiveness of VELF. VELF outperforms all the baselines on three datasets. Especially, the AUC improvements on 'New' and 'Infreq' test datasets are much more remarkable than 'All' test datasets. The results confirm the effectiveness of VELF to alleviate cold-start problem in CTR prediction.

Discussions. Firstly, with VELF, and similarly with DropoutNet and MWUF, the AUC improvements on 'Item' datasets are more significant than those on 'User' datasets. The reason is that in these three datasets, the data sparsity problem of items is more severe than that of users. Secondly, with DropoutNet, the performences on MovieLens datasets are much weaker than those on Taobao Display Ad and CIKM2019. To explain, recall that Taobao Display Ad and CIKM2019 have more abundant attributes for users and items which are relatively limited in MovieLens. Thus, content-based methods are more sensitive to the limitation of side information. Thirdly, on CIKM2019 dataset, AUC results of 'New user' and 'All' are equal. The reason is that users in test dataset and training dataset are non-overlapping by the default splitting settings.

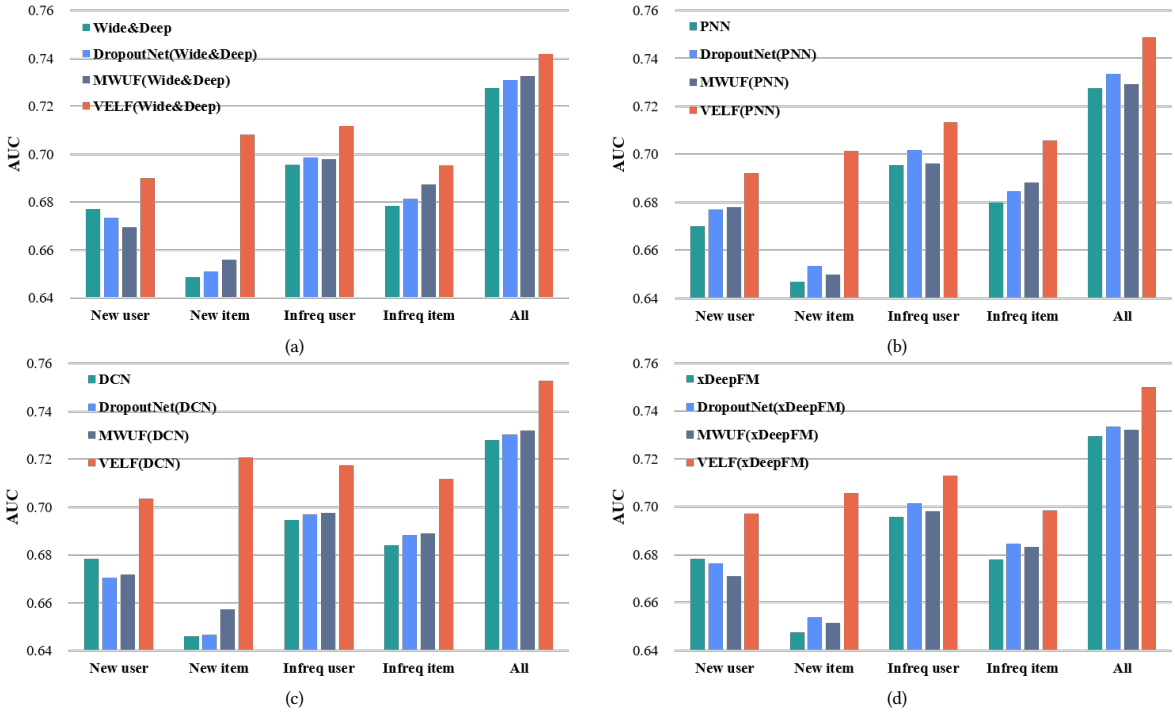

Figure 4: Performence on MovieLens-1M over four popular feature-crossing backbones: (a) Wide&Deep, (b) PNN, (c) DCN, (d) xDeepFM.

4.5 Generalization Experiments (RQ2)

In Section 4.4, we evaluate the effectiveness of VELF on the backbone of DeepFM. To further evaluate the generalization of VELF, we compare VELF with DropoutNet and MWUF in another four different popular network backbones, including Wide&Deep, PNN, DCN and xDeepFM. Experiments are conducted on the MovieLens-1M dataset.

The same as DropoutNet and MWUF, our VELF can be adapted to any network backbones by replacing the user and item embedding module. The experimental results are reported in Figure 4. The results indicate that VELF can constantly achieve the best performance with various base models.

4.6 Ablation Study (RQ3)

In this section, we demonstrate the advantages of our proposed distribution estimate, parameterized and regularized priors in VELF. We present an ablation study on MovieLens-1M dataset by evaluating several models based on DeepFM which is one of the lightest structures: (1) VELF: the overall framework; (2) VELF(Point): degenerate distribution estimate into point estimate by directly adopting μ_q^u as z^u and μ_q^i as z^i; (3) VELF(No-R): degenerate the parameterized and regularized priors into one layer parameterized priors without regularization by only keeping $p_{\phi_p^u}(z^u)$ and $p_{\phi_p^i}(z^i)$; (4) VELF(Fixed): degenerate the parameterized and regularized priors into one layer fixed normal priors. The mean AUC results over 5 runs are reported in Table 5.

Firstly, according to Table 4 and Table 5, VELF(Point) does not outperform DropoutNet much. This indicates the superiority of distribution estimate over point estimate. Secondly, VELF(No-R) outperforms VELF(Fixed) which confirms our claim that our proposed parameterized priors can improve the generalization ability. Thirdly, VELF is more effective than VELF(No-R). This demonstrates that constraining the parameterized priors to be close to a normal hyper-prior is helpful to further improve the generalization ability.

Table 5: Ablation study. The averaged AUC results over 5 runs are reported. Std ≈ 0.1%, extremely statistically significant under unpaired t-test.

Methods	New user	New item	Infreq user	Infreq item	All
VELF	0.7112	0.7106	0.7117	0.7009	0.7551
VELF(No-R)	0.6843	0.7037	0.7058	0.6935	0.7502
VELF(Fixed)	0.6723	0.6831	0.7070	0.6871	0.7402
VELF(Point)	0.6568	0.6508	0.6869	0.6723	0.731

5 CONCLUSION

In this paper, we propose a general Variational Embedding Learning Framework (VELF) to improve the generalization ability and robustness of embedding learning for cold-start users and Ads. VELF regards the embedding learning as a distribution estimate process, which means that embeddings are inferred from a series of shared distributions based on Bayesian inference. Thus the embeddings, especially the cold-start ones, can benefit from statistical strength among all the users and Ads. Besides, we develop a parameterized and regularized prior mechanism which can naturally utilizing the rich side information to further suppress overfitting. The embedding distributions and the discriminative CTR predicition

network parameters are learned end-to-end without strict requirements on extra training data or training stages. Experiments on several recommendation tasks show that CTR models with VELF can achieve better performances. Our future work will include the interactively modeling of user and Ad based on VELF and specific feature-crossing techniques under VELF.

REFERENCES

[1] Bahare Askari, Jaroslaw Szlichta, and Amirali Salehi-Abari. 2020. Joint variational autoencoders for recommendation with implicit feedback. *arXiv preprint arXiv:2008.07577* (2020).

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association* 112, 518 (2017), 859–877.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[4] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential scenario-specific meta learner for online recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2895–2904.

[5] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[6] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An Embedding Learning Framework for Numerical Features in CTR Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2910–2918.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[8] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph enhanced Embedding Neural Network for CTRPrediction. *arXiv preprint arXiv:2106.00314* (2021).

[9] Kilol Gupta, Mukund Yelahanka Raghuprasad, and Pankhuri Kumar. 2018. A hybrid variational autoencoder for collaborative filtering. *arXiv preprint arXiv:1808.01006* (2018).

[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[11] Diederik P Kingma and Max Welling. 2019. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691* (2019).

[12] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[13] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[14] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

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

[16] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. 2013. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. 283–292.

[17] Weiwen Liu, Ruiming Tang, Jiajin Li, Jinkai Yu, Huifeng Guo, Xiuqiang He, and Shengyu Zhang. 2018. Field-aware probabilistic embedding neural network for ctr prediction. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 412–416.

[18] Kaixiang Mo, Bo Liu, Lei Xiao, Yong Li, and Jie Jiang. 2015. Image feature learning for cold start problem in display advertising. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[19] Wentao Ouyang, Xiuwu Zhang, Shukui Ren, Li Li, Kun Zhang, Jinmei Luo, Zhaojie Liu, and Yanlong Du. 2021. Learning Graph Meta Embeddings for Cold-Start Ads in Click-Through Rate Prediction. *arXiv preprint arXiv:2105.08909* (2021).

[20] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 695–704.

[21] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[22] Sujoy Roy and Sharath Chandra Guntuku. 2016. Latent factor representations for cold-start video recommendation. In *Proceedings of the 10th ACM conference on recommender systems*. 99–106.

[23] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*. 880–887.

[24] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*. 89–96.

[25] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 253–260.

[26] Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman. 2011. Personalised rating prediction for new users using latent factor models. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*. 47–56.

[27] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. 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*. 528–536.

[28] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems.. In *NIPS*. 4957–4966.

[29] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[30] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. 2019. Variational few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1685–1694.

[31] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, and Xiaoming Li. 2015. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* 28, 5 (2015), 1147–1159.

[32] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[33] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. *arXiv preprint arXiv:2105.04790* (2021).

[34] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Warm Up Cold Item Embeddings for Cold-start Recommendation with Meta Scaling and Shifting Networks. *arXiv preprint arXiv:2105.04790* (2021).

A INFRA-COST ANALYSIS

We conducted experiments on a machine with a 10-core Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz and 100G memory, using tensorflow2.2.0. On Movielens dataset, 1.3G of memory was required by DeepFM when 1.4G of memory for DeepFM+VELF. The total computational complexity of DeepFM+VELF is less than 2x. DeepFM. Thanks to the high parallelism of CPU and tensorflow, the whole training time of DeepFM+VELF was only extended 10% compared to DeepFM under the same hardware constraints.