

Conversion Rate Prediction via Meta Learning in Small-Scale Recommendation Scenarios

Xiaofeng Pan^{*1}, Ming Li^{*1}, Jing Zhang², Keren Yu¹, Luping Wang¹, Hong Wen¹
Chengjun Mao¹ and Bo Cao¹

¹Alibaba Group, ²The University of Sydney

pxfvintage@163.com, {hongming.lm, keren.ykr, qinggan.wh, chengjun.mcj, zhizhao.cb}@alibaba-inc.com
jing.zhang1@sydney.edu.au, zjlxwlp8412@126.com

Abstract—Different from large-scale platforms such as Taobao and Amazon, developing CVR models in small-scale recommendation scenarios is more challenging due to the severe Data Distribution Fluctuation (DDF) issue. DDF prevents existing CVR models from being effective since 1) several months of data are needed to train CVR models sufficiently in small scenarios, leading to considerable distribution discrepancy between training and online serving; and 2) e-commerce promotions, which are being more prevalent and important to attract customers and boost sales, have much more significant impacts on small scenarios, leading to distribution uncertainty of the upcoming time period. In this work, we propose a novel CVR method named MetaCVR from a perspective of meta learning to address the DDF issue. Firstly, a base CVR model which consists of a Feature Representation Network (FRN) and output layers is elaborately designed and trained sufficiently with samples across months. Then we treat time periods with different data distributions as different occasions and obtain positive and negative prototypes for each occasion using the corresponding samples and the pre-trained FRN. Subsequently, a Distance Metric Network (DMN) is devised to calculate the distance metrics between each sample and all prototypes to facilitate mitigating the distribution uncertainty. At last, we develop an Ensemble Prediction Network (EPN) which incorporates the output of FRN and DMN to make the final CVR prediction. In this stage, we freeze the FRN and train the DMN and EPN with samples from recent time period, therefore effectively easing the distribution discrepancy. To the best of our knowledge, this is the first study of CVR prediction targeting the DDF issue in small-scale recommendation scenarios. Experimental results on real-world datasets validate the superiority of our MetaCVR and online A/B test also shows our model achieves impressive gains of 11.92% on PCVR and 8.64% on GMV.

Index Terms—Recommender System, Conversion Rate Prediction, Meta Learning, Data Distribution Fluctuation

I. INTRODUCTION

As an essential part of recommender system, Conversion Rate (CVR) prediction has been widely used in modern e-commerce and attracted huge attention from both academia and industry. Generally, CVR modeling methods employ similar techniques developed for Click-Through Rate (CTR) prediction, which use high-order interactions of features to improve their representation capacity [8], [31], [39] and leverage sequential user behaviors to model users in a dynamic manner [36], [41], [42]. However, due to label collection and dataset size problems, CVR modeling becomes quite different and challenging. The major difficulties of CVR modeling are

introduced by the well-known Sample Selection Bias (SSB) [40] and Data Sparsity (DS) [13] issues. Several studies have been carried out to tackle these challenges, *e.g.*, ESMM [16], ESM² [35] and HM³ [34].

However, most of the related works assume that data distribution of CVR training samples collected from different time periods is identical. For large-scale platforms such as Taobao and Amazon, this assumption is guaranteed by collecting training samples within a short time window, *e.g.*, a week or two, which is however far from enough in small scenarios to train well-performing deep CVR models. In small scenarios, usually several months of data are needed for CVR modeling. The large time span may result in discrepant data distributions between training and online serving, which hurts the generalization performance. Transfer learning [1], [2] can be applied to alleviate this problem by taking advantage of recent period of samples to fine-tune a pre-trained network that is trained with months of samples. However, the benefit of a pre-trained network decreases as the task on which the network was trained diverges from the target task [38].

Besides, in e-commerce, users' shopping decisions can be influenced by different *occasions* [30], which refer to time periods with different data distributions and are related to particular time or events. Especially, due to the intensification of e-commerce competition, promotions become quite frequent. For example, in our recommendation scenario, there exist two or more promotions every month with each promotion impacting about a week and resulting in various occasions. These promotions have remarkable impacts on the data distribution within a short time window. In such situations, CVR modeling is highly challenging since the distribution of the upcoming occasion is uncertain and training samples of each occasion become very scarce. As a result, the aforementioned transfer learning approaches [1], [2] are not effective in handling this challenge. As far as we know, there are no CVR methods focusing on tackling this problem.

We refer to the problems mentioned above as Data Distribution Fluctuation (DDF) issue, which widely exists in small-scale recommendation scenarios while remaining under-explored in CVR prediction. After a detailed analysis of the logs, we observe that purchases on different occasions show different patterns. For example, people tend to purchase items of intrinsic preferences on normal days while they tend to

^{*}Equal contribution.

engage with emerging hot items which are usually accompanied by attractive discounts during promotions. Moreover, user behaviors before, during and after promotions also vary a lot. Inspired by this, we consider that samples of each class on each occasion form a single prototype representation. Taking the distribution difference into consideration, we decompose the small scenario into four occasions, *i.e.*, Before-Promotion (BP), During-Promotion (DP), After-Promotion (AP) and Not-Promotion (NP). If all prototypes could be obtained and their distances to a sample to be predicted could be calculated, we can tackle the distribution uncertainty accordingly and thus address the DDF issue.

In this paper, we propose a novel CVR method named MetaCVR from a perspective of metric-based meta learning [9], [23] to embody the above idea. The meta-learning perspective is appealing as it provides an effective way to perform transfer learning across occasions, enabling us to cope with the further scarce training samples on each occasion after decomposition. Concretely, we design a base CVR model and train it sufficiently with samples across months despite the non-identical distribution. Discarding the output layers, the base CVR model is used as Feature Representation Network (FRN) which generates positive and negative prototypes for each occasion with corresponding samples. Then we elaborately devise a Distance Metric Network (DMN) to compute the distances between each sample and all prototype representations so that a sample’s priori conversion tendencies on different occasions are provided. At last, we develop an Ensemble Prediction Network (EPN) to incorporate the output of FRN and DMN and predict the final CVR score. When training the DMN and EPN, the FRN is frozen and samples from recent time period are adopted.

Our main contributions are summarized as follows:

- To the best of our knowledge, this is the first study of CVR prediction that handles the DDF issue in small-scale recommendation scenarios. We introduce the idea of decomposing the small scenario into different occasions according to their distribution difference.
- We propose a novel method from a perspective of meta learning to tackle the challenges introduced by the DDF issue, which models each occasion with prototype representations. Taking advantage of meta learning, our model achieves efficient knowledge sharing across occasions.
- Experiments on real-world datasets and online A/B test demonstrate the superiority of our model over representative methods. We also perform extensive analyses to confirm the effectiveness of our design.

II. RELATED WORK

A. CVR Prediction

CVR prediction is a key component of many online applications, such as search engines [18], recommender systems [33] and online advertising [15], [37]. Two critical issues make the CVR task quite challenging, *i.e.*, the SSB and DS issues. SSB refers to the issue that the sample space of CVR

prediction during training and inference can be systematically different, and DS refers to the fact that the amount of training samples for CVR prediction is much less than that for CTR prediction. For the SSB problem, Pan *et al.* [19] propose All Missing As Negative (AMAN) and treat unclicked samples as negative examples, which results in underestimated predictions. For the DS issue, oversampling samples of the rare class [32] is a widely used method. However, it is sensitive to sampling rates and not easy to achieve optimal results consistently. To relieve these problems, ESMM [16] models the “impression→click→purchase” path for the CVR task and trains the CVR model over the entire space. In this way, the SSB and DS issues can be mitigated and better performance can be achieved. Following the idea of ESMM and taking a step further, ESM² [35] and HM³ [34] elaborately design more complete post-click behavior decomposition and make use of purchase-related behaviors, leveraging the abundant supervisory signals to achieve better performance. In this paper, we focus on the DDF issue in CVR prediction, which has not been well-explored in existing methods. Actually, it should be treated as the primary problem to be solved in CVR modeling for small-scale scenarios, where most of the existing CVR methods can not perform well.

B. Meta Learning

Meta learning, or *learning to learn*, has been studied for a long time [21], [22], [25], and attracted increasing attention recently due to its potential in developing human-level artificial intelligence. Most existing meta learning approaches belong to either optimization-based or metric-based category. For optimization-based works, MAML [7], Meta-SGD [14] and TAML [10] aim to learn good model initialization so that new tasks can be learned well with a small amount of training samples, while another line of works focuses on learning an optimizer, including the LSTM-based meta-learner [20] and the weight-update mechanism with an external memory [17]. Metric-based meta learning is closely related to metric learning, which is usually initialized with a pre-trained model that projects inputs into a representation suitable for computing distance between query and support instances. Based on the representations and distance metrics, the whole model or a part of it can be trained further. For example, Siamese Network [11] compares new samples with existing ones in a learned metric space. Matching Network [28] and Prototypical Network [23] obtain the prediction of samples in query set by comparing the distance between the query set and the support set. Relation Network [24] shares the similar idea, but it replaces distance with a learnable relation module. Although meta learning has also been explored in recommender systems, *e.g.*, for algorithm selection [4], [5] and addressing the cold-start problem [6], [12], [26], [29], no attempt has been made to deal with the DDF issue in CVR prediction. In this study, we follow the idea of metric-based meta learning [23], [26] to fill the gap.

III. THE PROPOSED METHOD

A. Problem Setup

In CVR prediction, the model takes input as $(\mathbf{x}, y) \sim (X, Y)$, where \mathbf{x} is the features and $y \in \{0, 1\}$ is the conversion label. Specifically, the features for CVR prediction mainly consist of five parts. The first part is the user behavior sequence, which records user history of clicked/purchased items. The second part consists of the user features, including user profile (*e.g.*, age and gender) and statistic features from user history. The third part consists of the item features, *e.g.*, item id, category, brand and related statistic features. The fourth part is the interaction features of the target item and user, *e.g.*, clicks/purchases of the user in the category/shop during last 24 hours. The fifth part consists of context features, such as position, device, time information and occasion signals, in which occasion signals are sensitive to promotions.

The goal of CVR prediction is to learn a model f_θ parameterized by θ that minimizes the empirical risk:

$$\theta^* = \min_{\theta} E_{(\mathbf{x}, y) \sim (X, Y)} L(\mathbf{x}, y; f_\theta(\mathbf{x})), \quad (1)$$

where L is the loss function. In small-scale recommendation scenarios, the training dataset \mathcal{D} usually consists of samples collected from months of data in order to train CVR models sufficiently. Besides, an extra dataset \mathcal{D}_r is also constructed by collecting samples from recent time period.

B. Motivation

As illustrated in the introduction, we assume samples of each class on each occasion form a single prototype representation, *i.e.*, purchases driven by different occasions cluster around different patterns. Given a user-item pair, it is important to exploit the linkage between the representation of input features \mathbf{x} and occasion-driven purchase patterns to predict whether the conversion will occur. Since different patterns coexist on each occasion with different impacts, simply distinguishing the source occasion of samples by occasion signals in input features would not deliver good performance. Instead, we formulate the CVR prediction in small scenarios as:

$$f_\theta(\mathbf{x}) = g(\mathcal{F}(\mathbf{x}), \{d(\mathcal{F}(\mathbf{x}), \mathbf{p}_{occ}^{cls})\}), \quad (2)$$

where $\mathcal{F}(\mathbf{x})$ denotes feature representation of \mathbf{x} and $d(\cdot)$ denotes the distance metric function. With $cls \in \{+, -\}$ and $occ \in \{BP, DP, AP, NP\}$, \mathbf{p}_{occ}^{cls} denotes the pattern of class cls on occasion occ , and $g(\cdot)$ is the final prediction function. Symbol $+$ denotes the positive class while $-$ denotes the negative. In this paper, we refer to the patterns as prototypes and implement $\mathcal{F}(\cdot)$, $d(\cdot)$ and $g(\cdot)$ as FRN, DMN and EPN respectively, which are shown in Figure 1 and will be detailed in the following sections.

C. Base CVR Model

Since the performance of our proposed method greatly relies on $\mathcal{F}(\mathbf{x})$, it is of great importance to design a base CVR model, which can generate a high-quality feature representation and predict an accurate CVR score accordingly.

With output layers discarded, the base CVR model can serve as the FRN which projects input features into an implicit representation space. The details are presented as follows.

Firstly, a shared embedding layer is adopted to handle the input features \mathbf{x} as mentioned in Section III-A. They can be further grouped into two kinds of features: categorical feature and numerical feature. We discrete the numerical features based on their boundary values, transforming them into the categorical type. Then each categorical feature is encoded as a one-hot vector. Due to the sparseness nature of one-hot encoding, we apply linear fully connected layers to obtain low dimensional embedding. After embedding, e^u , e^i , e^{ui} and e^c denote user features, item features, user-item interaction features and context features respectively. Then the embedding of user behavior sequence is formed with item embeddings in the sequence, *i.e.*, $e^{ubs} = \{e_1^i, \dots, e_t^i\}$ where e_t^i denotes the item embedding of t^{th} user behavior and t is the sequence length.

Subsequently, MainNet is devised to model the target item and user. Three kinds of attention [27] weights are calculated based on the embedding of user behavior sequence since it contains rich information about user interest. First, we use a multi-head self-attention network to model user preference from multiple views of interest. For self-attention, Query, Key, and Value all refer to e^{ubs} and $\hat{e}^{ubs} = \{\hat{e}_1^i, \dots, \hat{e}_t^i\}$ is the output. On top of the self-attention network, user attention and target attention are performed in parallel. With e^u as Query and \hat{e}^{ubs} as Key and Value, user attention s^u is calculated to mine personalized information and suppress noisy behavior. Similarly, with e^i as Query and \hat{e}^{ubs} as Key and Value, target attention s^i is applied to activate historical interests related to the target item. At last, e^u , e^i , e^{ui} , s^i and s^u are concatenated and fed into a Multi-Layer Perception (MLP), generating \mathbf{h}^M as the output of MainNet. It can be observed that different users in different context usually behave differently even to similar items. Therefore, we concatenate e^u with e^c and feed them into another MLP (BiasNet) to model the bias. \mathbf{h}^B denotes the output of the BiasNet.

Finally, we obtain the output of FRN and feed it to output layers to predict base CVR scores. We adopt the widely-used logloss as loss function to train the base CVR model, *i.e.*,

$$\begin{aligned} \hat{y}_b &= f(\mathcal{F}(\mathbf{x})) = f(\text{Norm}([\mathbf{h}^M; \mathbf{h}^B])), \\ l_b &= -\frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, y) \in \mathcal{D}} (y \log \hat{y}_b + (1 - y) \log(1 - \hat{y}_b)), \end{aligned} \quad (3)$$

where $[\cdot]$ refers to concatenation of vectors, and $f(\cdot)$ is a ranking function implemented as a 3-layer MLP of which the last layer uses Sigmoid as activation function while the other layers use ReLU.

D. Prototype Representations

Considering the differences of user behaviors on different occasions, we build the support set for each occasion via picking a day of this occasion in \mathcal{D}_r and splitting its samples into 2 subsets, *i.e.*, positive support set and negative support

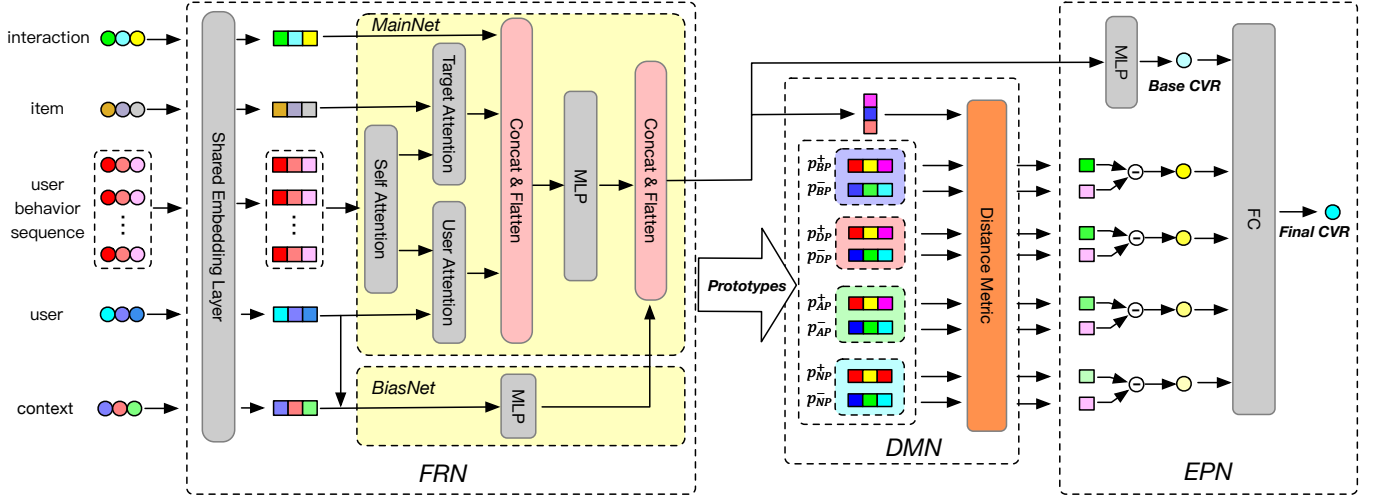


Fig. 1. Framework of the proposed MetaCVR method. It contains a Feature Representation Network (FRN), a Distance Metric Network (DMN), and an Ensemble Prediction Network (EPN).

set. The positive support set includes all purchase samples of the day while the negative includes the rest of clicked samples. Then we take advantage of the pre-trained FRN to map the input into a representation space and calculate the class's prototype as the mean feature of its support set, *i.e.*,

$$\begin{aligned} V_{occ}^{cls} &= \{\mathcal{F}(\mathbf{x}_k) \mid \mathbf{x}_k \in S_{occ}^{cls}\}, \\ \mathbf{p}_{occ}^{cls} &= \text{Norm}(\text{Mean}(V_{occ}^{cls})), \end{aligned} \quad (4)$$

where S_{occ}^{cls} denotes the support set of class cls on occasion occ , and \mathbf{x}_k denotes input features of k th sample in S_{occ}^{cls} . In this way, we obtain 4 pairs of prototypes, *i.e.*, $\{\{\mathbf{p}_{BP}^+, \mathbf{p}_{BP}^-\}, \{\mathbf{p}_{DP}^+, \mathbf{p}_{DP}^-\}, \{\mathbf{p}_{AP}^+, \mathbf{p}_{AP}^-\}, \{\mathbf{p}_{NP}^+, \mathbf{p}_{NP}^-\}\}$.

E. Distance Metric Network

In metric-based meta learning, the choice of distance metric is crucial. In this paper, since the representation space of FRN is highly non-linear, it may not be suitable to choose fixed linear distance metrics such as cosine distance and Euclidean distance adopted in [23], [28]. We consider that a learnable distance metric can be a more generalizable solution and propose a trainable Space Projection Distance Metric (SPDM) which is formulated as follows:

$$d(\mathcal{F}(\mathbf{x}), \mathbf{p}_{occ}^{cls}) = \mathcal{F}(\mathbf{x})^T \mathbf{W}_{occ} \mathbf{p}_{occ}^{cls} + b_{occ}, \quad (5)$$

where \mathbf{W}_{occ} is a trainable projection matrix and b_{occ} is a trainable bias scalar. It is worth mentioning that cosine distance is a special case of SPDM when \mathbf{W}_{occ} is an identity matrix.

Another style of trainable distance metric is similar to Relation Network [24], which aims to learn the relation between query sample and support sets as a transferrable deep metric. In this paper, we also borrow this idea and propose a Neural Network based Distance Metric (NNDM), *i.e.*,

$$d(\mathcal{F}(\mathbf{x}), \mathbf{p}_{occ}^{cls}) = \mathbf{W}_{occ} [\mathcal{F}(\mathbf{x}); \mathbf{p}_{occ}^{cls}] + b_{occ}, \quad (6)$$

F. Ensemble Prediction Network

From SPDM or NNDM, we can obtain four pairs of distance metrics, *i.e.*, $\{\{d_{BP}^+, d_{BP}^-\}, \{d_{DP}^+, d_{DP}^-\}, \{d_{AP}^+, d_{AP}^-\}, \{d_{NP}^+, d_{NP}^-\}\}$. In most of the existing metric-based works, classification of a query sample is then performed by simply finding its nearest prototype, which is not directly applicable for CVR prediction since prototypes do not maintain fine-grained personalized information after mean pooling, which however is essential for a well-performing CVR model. Alternatively, we incorporate these distance metrics with the output of FRN in an ensemble approach, *i.e.*,

$$\hat{y} = f_e(s_b, \{s_{occ}\}), s_b = f(\mathcal{F}(\mathbf{x})), s_{occ} = d_{occ}^+ - d_{occ}^-, \quad (7)$$

where s_b denotes CVR prediction of the base model and $f(\cdot)$ refers to the ranking function in Eq. (3). s_{occ} represents how likely the purchase would happen on occasion occ . The final CVR score \hat{y} is predicted by a fully connected layer, *i.e.*, $f_e(\cdot)$, with $s_{NP}, s_{BP}, s_{DP}, s_{AP}$ and s_b as input and Sigmoid as activation function. Similarly to Eq. (3), logloss is adopted to train the whole model, which is defined as:

$$L = -\frac{1}{|\mathcal{D}_r|} \sum_{(x,y) \in \mathcal{D}_r} (y \log \hat{y} + (1-y) \log(1-\hat{y})). \quad (8)$$

Note that in this stage, we only train the parameters of DMN and EPN on \mathcal{D}_r by stopping the gradient propagation to FRN. Algorithm 1 shows the detailed two-stage training procedure.

IV. EXPERIMENTS

In this section, we conduct a series of experiments to investigate the following research questions:

RQ1 How does our proposed MetaCVR method perform compared to the state-of-the-art (SOTA) models for CVR prediction in the small-scale recommendation scenario?

RQ2 How do different distance metrics affect the performance of MetaCVR?

Algorithm 1 Training Procedure of the proposed MetaCVR**Input:** training datasets: \mathcal{D} and \mathcal{D}_r ;**Output:** optimal θ (parameters in Equation 1);

```

1: [Training Stage 1]
2: randomly initialize  $\theta_1$  (parameters of FRN)
3: randomly initialize  $\theta_2$  (parameters of  $f(\cdot)$  in Equation 3)
4: while not converge do
5:   sample mini-batches from  $\mathcal{D}$ 
6:   compute losses according to Equation 3
7:   update  $\theta_1$  and  $\theta_2$  via Adagrad
8: end while
9: [Training Stage 2]
10: for  $occ$  in  $\{BP, DP, AP, NP\}$  do
11:   build support sets  $S_{occ}^+$  and  $S_{occ}^-$  from  $\mathcal{D}_r$ 
12:   compute prototypes  $p_{occ}^+$  and  $p_{occ}^-$  according to Equation 4
13: end for
14: randomly initialize  $\theta_3$  (parameters of DMN)
15: randomly initialize  $\theta_4$  (parameter of  $f_e(\cdot)$  in Equation 7)
16: while not converge do
17:   sample mini-batches from  $\mathcal{D}_r$ 
18:   compute losses according to Equation 8
19:   freeze  $\theta_1$  and update  $\theta_2, \theta_3$  and  $\theta_4$  via Adagrad
20: end while

```

RQ3 Why is our MetaCVR method effective in small-scale recommendation scenarios?

A. Experimental Setup

1) **Datasets:** We establish the whole dataset by collecting the users' interaction logs¹ from a small-scale recommendation scenario of our online e-commerce platform, where promotions are highly frequent and have a considerable impact on the recommender system. Logs are sampled from 2021/01/15 to 2021/03/31, including 3 monthly promotions (21st of each month) and promotions for Spring Festival, Valentine's Day and Women's Day. We split the entire dataset into non-overlapped training set \mathcal{D} (2021/01/15-2021/03/15) and validation set \mathcal{D}_v according to the timestamp of the prediction behavior, effectively avoiding feature leakage. In this way, the training set is about 80% of the whole dataset and the left 20% of data is used as the validation set. From the training set, we further build \mathcal{D}_r with data from 2021/03/01 to 2021/03/15. The detailed statistics of these datasets are summarized in Table I. All the data we use have been anonymously processed by the log system and user's information is protected.

2) **Evaluation Metrics:** Area under ROC curve (AUC) is used as the offline evaluation metric. For online A/B testing, we choose PCVR = $p(\text{conversion}|\text{click}, \text{impression})$ and GMV (Gross Merchandise Volume), which are widely adopted in industrial recommender systems for evaluating online performance. Improving PCVR and GMV simultaneously implies more accurate recommendations and business revenue growth.

¹To the extent of our knowledge, there are no public datasets suited for the CVR prediction task in settings of this paper.

TABLE I
STATISTICS OF THE ESTABLISHED DATASETS.

#Dataset	#Users	#Items	#Exposures	#Clicks	#Purchases
\mathcal{D}	4.68M	0.71M	461.48M	42.32M	413K
\mathcal{D}_r	2.62M	0.68M	121.37M	12.38M	141K
\mathcal{D}_v	2.12M	0.68M	119.21M	11.17M	113K

3) **Comparison Methods:** The representative comparison methods are described as follows. 1) **XGBoost** [3] is a tree-based model which can produce competitive, robust and interpretable results for CVR prediction and is especially suitable when training samples are not sufficient for training deep models. 2) **DCN** [31] applies feature crossing at each layer. The advanced ability to capture high-order feature interactions makes DCN a better choice than DNN for CVR prediction. 3) **BASE** refers to the base CVR model proposed in Section III-C, which models sequential user behaviors via the attention mechanism. 4) **BASE-F** is the same model as BASE except that it adopts a two-stage training process to relieve the distribution discrepancy caused by the large time span. 5) **ESMM** [16] mitigates the SSB and DS issues by modeling CVR on user sequential path "impression→click→purchase". 6) **ESM²** [35] extends ESMM and models purchase-related post-click behaviors in a unified multi-task learning framework.

Features for all the above methods are the same except that XGBoost and DCN discard user behavior sequence. Besides, since XGBoost requires less training data than deep models, we train two XGBoost models in this work, one with \mathcal{D} and the other with samples of the last 20 days in \mathcal{D} . The latter is used as a comparison model since it outperforms the former. For a fair comparison, the BASE structure is adopted for each task in ESMM and ESM². HM³ is not compared since micro behaviors are not available in our scenario.

4) **Implementation Details:** For the XGBoost model, the number and depth of trees, minimum instance numbers for splitting a node, sampling rate of the training set and sampling rate of features for each iteration are set to 70, 5, 10, 0.6 and 0.6, respectively. All the other deep models are implemented in distributed Tensorflow 1.4. During training, we use 2 parameter servers and 3 Nvidia Tesla V100 16GB GPUs. Item ID, category ID and brand ID have an embedding size of 32 while 8 for the other categorical features. We use 8-head attention structures with a hidden size of 128. Adagrad optimizer with a learning rate of 0.01 and a mini-batch size of 256 are used for training. We report the results of each method under its empirically optimal hyper-parameters settings.

B. Experimental Results: RQ1

The offline and online comparison results are presented in Table II. For offline evaluation, all experiments are repeated 3 times on \mathcal{D}_v . For online A/B testing, XGBoost was used as the baseline and the other models were tested in turn since the online traffic was not enough for testing all models simultaneously. Time for A/B testing of each model covered all occasions mentioned in this paper. The major observations are summarized as follows:

TABLE II
OFFLINE AND ONLINE RESULTS. BOLD: BEST. UNDERLINE: RUNNER-UP.

Model	Offline	Online	
	AUC (mean \pm std.)	PCVR Gain	GMV Gain
XGBoost	0.8264 \pm 0.00067	0%	0%
DCN	0.8174 \pm 0.00074	-	-
BASE	0.8233 \pm 0.00257	-5.22%	-6.58%
BASE-F	0.8259 \pm 0.00187	-1.01%	-1.42%
ESMM	0.8262 \pm 0.00163	-0.83%	-0.61%
ESM ²	0.8278 \pm 0.00151	<u>+2.36%</u>	<u>+3.02%</u>
MetaCVR	0.8304\pm0.00156	+11.92%	+8.64%

TABLE III
COMPARISON OF DISTANCE METRIC METHODS.

Distance Metric Method	AUC (mean \pm std.)
Cosine	0.8269 \pm 0.00103
Euclidean	0.8276 \pm 0.00124
NNDM	0.8296 \pm 0.00161
SPDM	0.8304 \pm 0.00156

- XGBoost is a strong baseline for our small scenario, which ranks third and outperforms all the other deep models except for ESM² model and our MetaCVR. It requires fewer training samples and thus alleviates the distribution discrepancy caused by the large time span.
- The elaborately designed BASE model outperforms the DCN model, validating the effectiveness of our sequential modeling method. Moreover, the BASE-F further improves the performance by fine-tuning the BASE model using recent samples, implying impacts of the distribution discrepancy do matter. However, it still lags behind XGBoost. A possible reason is that increasing model parameters imposes difficulties in training in small scenarios.
- ESMM and ESM² deliver better performance owing to user behavior decomposition. With auxiliary tasks, the SSB and DS issues are mitigated and models are trained more sufficiently. ESMM is comparable to XGBoost while ESM² outperforms XGBoost by adopting more purchase-related signals, becoming the runner-up method.
- Our MetaCVR model yields the best performance on both offline and online settings by addressing the DDF issue effectively via meta learning, outperforming the runner-up methods by a large margin.

C. Impact of distance metrics: RQ2

Then, we explore the impacts of distance metrics in the DMN module. Concretely, we train four MetaCVR models using cosine distance, Euclidean distance, NNDM and SPDM in the DMN respectively. The results on \mathcal{D}_v are presented in Table III. We can find that performance achieved by different distance metric methods varies significantly, which confirms that the choice of distance metric is crucial. SPDM and NNDM outperform cosine and Euclidean obviously, validating our idea that learnable distance metrics are better solutions especially when the characteristics of representation space are unknown. Besides, our proposed SPDM outperforms NNDM and achieves the best performance. It is because the induc-

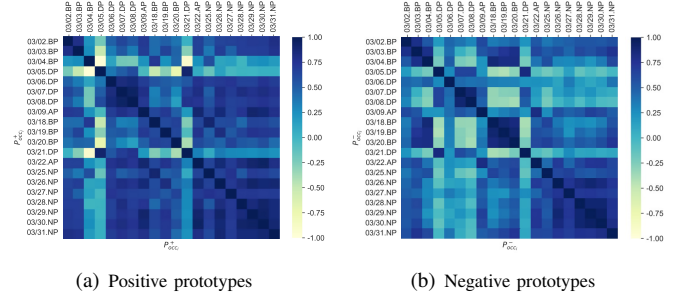


Fig. 2. Cosine similarity of prototype representations in different occasions.

tive bias of SPDM is simpler than NNDM when measuring distance metrics, which is beneficial in the context of limited data.

D. Effectiveness analysis: RQ3

As summarized in Section IV-B, BASE-F performs better than BASE, and more impressive improvements are achieved when MetaCVR takes advantage of prototypes. Intuitively, BASE-F tackles the distribution discrepancy caused by the large time span, while MetaCVR partially employs this idea and further tackles the distribution uncertainty introduced by promotions, which we consider as the main reason for performance improvements. Taking a step further, we investigate the effectiveness of prototypes. First, days of different occasions are picked from both training set and validation set. Then prototype representations of each chosen day are obtained according to Section III-D and we compute cosine similarity of positive and negative prototypes respectively. As shown in Figure 2, prototypes of the same occasion have high similarities while prototypes of different occasions share low similarities. For example, the negative prototypes in Figure 2(b) of an occasion (03/02-03/04) have high similarities with that of the same occasion from another promotion (03/18-03/20) while prototypes of different occasions from the same promotion (03/18-03/22) share low similarities. The visualization empirically confirms our assumption that prototypes of different occasions can distinguish from each other. With the learnable DMN, prototypes can be used to provide a sample's priori conversion tendencies on different occasions and thus help to handle the distribution uncertainty.

V. CONCLUSION

In this paper, we investigate the severe DDF issue in small-scale recommendation scenarios and propose a novel CVR prediction method named MetaCVR. It leverages the idea of metric-based meta learning to cope with the distribution uncertainty caused by frequent promotions and delivers promising transfer learning performance across occasions. Experiments on both offline datasets and online A/B test show MetaCVR significantly outperforms representative models. In the future, we intend to combine our method with the idea of ESM² for further improvements, which is motivated by experimental results in Table II.

REFERENCES

- [1] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.
- [2] Rich Caruana. Learning many related tasks at the same time with backpropagation. In *Advances in neural information processing systems*, pages 657–664, 1995.
- [3] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [4] Tiago Cunha, Carlos Soares, and André CPLF de Carvalho. Selecting collaborative filtering algorithms using metalearning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 393–409. Springer, 2016.
- [5] Tiago Cunha, Carlos Soares, and André CPLF de Carvalho. Metalearning and recommender systems: A literature review and empirical study on the algorithm selection problem for collaborative filtering. *Information Sciences*, 423:128–144, 2018.
- [6] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. Sequential scenario-specific meta learner for online recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2895–2904, 2019.
- [7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.
- [8] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.
- [9] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.
- [10] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019.
- [11] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.
- [12] Hyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1073–1082, 2019.
- [13] Kuang-chih Lee, Burak Orten, Ali Dasdan, and Wentong Li. Estimating conversion rate in display advertising from past performance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 768–776. ACM, 2012.
- [14] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.
- [15] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD’17*, pages 1–9, 2017.
- [16] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1137–1140. ACM, 2018.
- [17] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.
- [18] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 596–605, 2018.
- [19] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*, pages 502–511. IEEE, 2008.
- [20] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.
- [21] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.
- [22] Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1):105–130, 1997.
- [23] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.
- [24] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.
- [25] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.
- [26] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. 2017.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [28] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.
- [29] Huiwei Wang and Yong Zhao. M2e: Meta-learning embedding ensemble for cold-start recommendation. *IEEE Access*, 8:165757–165768, 2020.
- [30] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. Time to shop for valentine’s day: Shopping occasions and sequential recommendation in e-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 645–653, 2020.
- [31] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD’17*, pages 1–7, 2017.
- [32] Gary M Weiss. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1):7–19, 2004.
- [33] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. Multi-level deep cascade trees for conversion rate prediction in recommendation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- [34] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. Hierarchically modeling micro and macro behaviors via multi-task learning for conversion rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.
- [35] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2377–2386, 2020.
- [36] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2265–2268, 2020.
- [37] Hongxia Yang, Quan Lu, Angus Xianen Qiu, and Chun Han. Large scale cvr prediction through dynamic transfer learning of global and local features. In *Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pages 103–119, 2016.
- [38] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.
- [39] Feng Yu, Zhaocheng Liu, Qiang Liu, Haoli Zhang, Shu Wu, and Liang Wang. Deep interaction machine: A simple but effective model for high-order feature interactions. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2285–2288, 2020.
- [40] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the international conference on Machine learning*, 2004.
- [41] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for

- click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5941–5948, 2019.
- [42] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.