Task Adaptive Multi-learner Network for Joint CTR and CVR Estimation

Xiaofan Liu, Qinglin Jia, Chuhan Wu, Jingjie Li, Quanyu Dai, Lin Bo, Rui Zhang, Ruiming Tang

1Noah’s Ark Lab, Huawei 2Beijing University of Posts and Telecommunications 3Renmin University of China 4ruizh@g.com

1jiaqinglin2,wuchuhan,lijingj1,daiquanyu,tangruiming@huawei.com,liuxiaofan@bupt.edu.cn
bolin20@ruc.edu.cn,rayteam@yeah.net

ABSTRACT
CTR and CVR are critical factors in personalized applications, and many methods jointly estimate them via multi-task learning to alleviate the ultra-sparsity of conversion behaviors. However, it is still difficult to predict CVR accurately and robustly due to the limited and even biased knowledge extracted by the single model tower optimized on insufficient conversion samples. In this paper, we propose a task adaptive multi-learner (TAML) framework for joint CTR and CVR prediction. We design a hierarchical task adaptive knowledge representation module with different experts to capture knowledge in different granularities, which can effectively exploit the commonalities between CTR and CVR estimation tasks while keeping their unique characteristics. We apply multiple learners to extract data knowledge from various views and fuse their predictions to obtain accurate and robust scores. To facilitate knowledge sharing across learners, we further perform self-distillation that uses the fused scores to teach different learners. Thorough offline and online experiments show the superiority of TAML in different Ad ranking tasks, and we have deployed it in Huawei’s online advertising platform to serve the main traffic.

CCS CONCEPTS
• Information systems → Computational advertising; • Computing methodologies → Multi-task learning.

KEYWORDS
Multi-task Learning, Recommender System, CVR Estimation

1 INTRODUCTION
Online advertising is essential to customer acquisition, which typically follows a sequential event pattern of “impression → click → conversion” [22]. For example, in the app promotion scenario, users may first click an Ad if they are attracted by its content, and then decide whether to trigger a conversion by installing the corresponding App or submitting registration forms [1]. To target users’ interest and increase advertisers’ return on investment (ROI), click-through rate (CTR) prediction and post-click conversion rate (CVR) prediction are essential tasks for personalized Ad ranking systems [4, 10, 17, 19], both of which have decisive impacts on advertisers’ bid under various mainstream pricing rules like optimized cost-per-click (OCPC) [12].

Due to the cascaded process of customer acquisition, the amount of positive feedback is progressively reduced (left Fig. 1), which leads to a severe sparsity problem of conversion events and thereby differentiates the characteristics of CTR and CVR model learning [10, 14]. However, conventional methods usually separately estimate CTR and CVR with similar techniques (e.g., factorization machines [5, 23] and deep neural networks [DNN] [3, 7]), which may suffer from the sparsity of conversion samples [10, 15, 19]. Thus, there has been an increasing trend in industrial applications to jointly estimate CTR and CVR via multi-task learning (MTL) [8–10, 20, 21]. For example, Ma et al. [10] propose ESMM to optimize both CTR and CTCVR (click-through conversion rate) tasks on post-view samples, where the CTCVR score is the multiplication of predicted CTR and CVR scores. Ma et al. [9] propose to use shared experts controlled by gating mechanisms below task-specific networks, which aims to model the commonalities among tasks. However, due to the ultra-sparsity of conversion events, the knowledge learned by these methods for CVR estimation may still be rather limited and even biased [2, 13]. Concerning the diverse CTCVRs of different items
In this paper, we propose a Task Adaptive Multi-learner (TAML) framework for joint CTR and CVR estimation, which employs a hierarchical task adaptive knowledge representation module with different experts to capture multi-grained knowledge, and uses multiple learners to exploit and exchange knowledge from multiple views to facilitate accurate and robust prediction. Specifically, the hierarchical task adaptive knowledge representation module uses three groups of experts to respectively capture task-agnostic, task-specific, and even finer-grained learner-specific knowledge. On the top of these experts, each task has multiple learners to adapt to its characteristics and make predictions, which are further synthesized into the final output. To facilitate knowledge exchange among learners, we apply self-distillation[18, 23] by using the fused output as a virtual teacher to teach them to make accurate and robust predictions. Both online and offline experiments show the advantage of TAML over baselines, and it has been serving the main traffic of Huawei’s online advertising system for mobile app promotion.

2 METHODOLOGY

2.1 Overall Framework

Suppose the input features of the model are $X$, which are looked up by a shared embedding layer to get the feature embedding vector $x$. Then it is input to a hierarchical task adaptive knowledge representation (HTAKR) module, which extracts knowledge from the feature embedding in three granularities, including a general level, a task-specific level, and a finer-grained learner-specific level. The outputs from this module will be aggregated by a set of adaptive gates, which adaptively control the importance of the three types of knowledge according to the feature embedding. Each gate corresponds to a task-specific learner, which further encodes knowledge from different views and makes predictions for its task. The prediction scores from multiple learners are aggregated into a unified score as the final output of its task, which also serves as the virtual teacher to teach learners of this task via self-distillation to exchange cross-learner knowledge. We finally multiply the probability results of CTR and CVR tasks to get the CTCVR score $\hat{p}_{ctcvr}$.

2.2 HTAKR Module

To fully model task-agnostic and task-specific knowledge, we design a hierarchical task-adaptive knowledge representation module on the top of the embedding layer. It includes general experts to learn universal knowledge across tasks, task-specific experts to learn task-sensitive knowledge, and learner-specific experts to further cover task-aware knowledge in different aspects. To adaptively fuse the knowledge extracted by the three groups of experts, an adaptive gate is used to generate the input embedding for each learner from the outputs of general experts and experts of the corresponding task and learner, as shown in right Fig. 2. Let $l$ represent the $l$-th learner and $t$ represent the task $t \in \{ctr, cor\}$, the input of each learner is denoted as $f_l^t(x)$. The adaptive gate uses a single-layer feed-forward network with Softmax function to assign weights to each expert and aggregate the inputs as follows:

$$g_l^t(x) = \text{Softmax} \left( W_l^t x \right), \quad f_l^t(x) = \sum_{i=1}^{k} g_l^t(x) e_i^t(x), \quad (1)$$

where $W_l^t$ are parameters of the $l$-th learner, $k$ is the total number of experts, $e_i^t(x)$ is the concatenation of the embeddings generated by the general experts, task $t$’s experts, and $l$-th learner’s expert.
2.3 Multi-Learner Network with Self-Distillation

The input of each learner $f_l(x)$ first passes through a multi-layer perceptron and gets the output logits, i.e., $logit_l = MLP(f_l(x))$. Then the output logits of each learner are aggregated by average as $logit_t = \frac{1}{n} \sum_{i=1}^{n} logit_l$, where logit_t represents the ensemble results in the task $t$ and $n$ is the total number of learners in this task. The ensemble results are normalized by the sigmoid function, which is denoted as $p_t$. To encourage knowledge sharing among learners of each task, we apply self-distillation to the multi-learner network. We set the fused results as the teacher and each learner as the student. Denote the normalized output of each learner as $\hat{p}_l$, the self-distillation loss functions in different tasks are formulated as follows:

$$L_{self-kdcv} = -\sum_{i=1}^{n} [p_{ctr} \log(\hat{p}_{ctr}) + (1 - p_{ctr})(1 - \log(\hat{p}_{ctr}))],$$

$$L_{self-kdcv} = -\sum_{i=1}^{m} [p_{cvr} \log(\hat{p}_{cvr}) + (1 - p_{cvr})(1 - \log(\hat{p}_{cvr}))].$$

2.4 Joint Loss Optimization for TAML

Following [10], we use the cross-entropy loss function to supervise the CTR and CTCVR tasks as follows:

$$L_{ctr} = -\frac{1}{N} \sum_{i=1}^{N} [y_{ctr} \log(\hat{p}_{ctr}) + (1 - y_{ctr}) \log(1 - \hat{p}_{ctr})].$$

$$L_{ctcvr} = -\frac{1}{N} \sum_{i=1}^{N} [y_{ctcvr} \log(\hat{p}_{ctcvr}) + (1 - y_{ctcvr}) \log(1 - \hat{p}_{ctcvr})].$$

where $L_{ctr}$ and $L_{ctcvr}$ are the loss functions of the CTR and CTCVR tasks, respectively. The final loss function $L$ of the TAML framework consists of four parts, including the loss of two supervised tasks and the loss of two self-distillation auxiliary tasks:

$$L = L_{ctr} + L_{ctcvr} + \alpha L_{self-kdcv} + \beta L_{self-kdcv}$$

where $\alpha$ and $\beta$ are hyper-parameters, which are the weights of the self-distillation loss in CTR and CVR tasks.

3 EXPERIMENTS

3.1 Datasets

We use two datasets in the experiments. The first one is a public dataset named Alibaba Click and Conversion Prediction (Ali-CCP). It is collected from real-world traffic logs of the recommender systems in the Taobao platform. Following [16], we filter the features whose frequencies are less than 10 and divide 10% of the samples in the original training set for validation. The second one is a proprietary dataset (named Industrial) collected from Huawei's advertising platform. It contains 8 consecutive days of logged data from a browser search application. The feature set of this dataset is comprised of 12 fields, including user attributes, item information, and corresponding context. We use samples on days 1-7 for training and day 8 for test. The details of both datasets are listed in Table 1.

3.2 Experimental Settings

In our experiments, we optimize the hyperparameters of baselines and our methods by grid search. We set up a four-layer deep neural network for all MLP-based methods, and the numbers of hidden units in each layer are [192, 64, 32, 1]. We use three experts in expert-based models, each of which is a one-layer perceptron with 64 hidden units. In our method, we use two learners for each task, and each learner is a three-layer perceptron with 64, 32, and 1 hidden units. The weights of self-distillation loss $\alpha$ and $\beta$ are set to 0.1. For model training, Adam [6] is the optimizer with a learning rate of 1e-3, and the batch size is 2000. We use L2 regularization with the strength of 1e-6. ReLU is the activation function. In all offline experiments, we use ROC AUC of CTR, CVR, CTCVR tasks as the evaluation metric. We run five rounds of experiments for each method and record the average results.

3.3 Offline Evaluation

We conduct extensive experiments on the Ali-CCP and industrial datasets by comparing TAML with the following baselines: (1) MLP: using separate multi-layer perceptrons to predict the target for each task; (2) MMoE: using multiple shared experts to extract common knowledge; (3) PLE: using both task-specific experts and shared experts to achieve progressive knowledge extraction; (4) ESSM: joint estimation of different tasks via probability multiplication; (5) AITM: transferring information from previous tasks to backward tasks and using a calibrator to constrain the prediction. The evaluation results are shown in Table 2, from which we have the following observations. In the two tasks of CVR and CTCVR, MLP shows low performance, which is due to the sparsity of conversion samples. Besides, MLP does not use the multi-task learning method to learn the CTR task, which leads to insufficient learning in the discrimination of impression samples in the testing data. PLE is better than MMoE in most tasks since the task-specific experts in PLE can independently assist in each task learning. PLE can learn the commonalities and differences between CTR and CVR.

---

### Table 1: Statistics of the two datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#User</th>
<th>#Item</th>
<th>#Impression</th>
<th>#Click</th>
<th>#Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ali-CCP</td>
<td>0.4M</td>
<td>4.3M</td>
<td>84M</td>
<td>3.4M</td>
<td>18k</td>
</tr>
<tr>
<td>Industrial</td>
<td>0.5M</td>
<td>67.6k</td>
<td>736M</td>
<td>3.7M</td>
<td>1.8M</td>
</tr>
</tbody>
</table>

### Table 2: Model AUC on the Ali-CCP and Industrial datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Ali-CCP Dataset</th>
<th>Industrial Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTR</td>
<td>CVR</td>
<td>CTCVR</td>
</tr>
<tr>
<td>MLP</td>
<td>0.6034</td>
<td>0.6519</td>
</tr>
<tr>
<td>MMoE</td>
<td>0.6026</td>
<td>0.6703</td>
</tr>
<tr>
<td>PLE</td>
<td>0.6037</td>
<td>0.6712</td>
</tr>
<tr>
<td>ESSM</td>
<td>0.6030</td>
<td>0.6706</td>
</tr>
<tr>
<td>AITM</td>
<td>0.6035</td>
<td>0.6693</td>
</tr>
<tr>
<td>TAML</td>
<td>0.6049</td>
<td>0.6813</td>
</tr>
</tbody>
</table>

---

1https://tianchi.aliyun.com/datalab/datSet.html?dataId=408
prediction and improves overall performance. These two methods achieve improvement to some degree, but they have not taken advantage of the sequential relationship between CTR and CTCVR tasks. ESMM implicitly models $p_{ctcvr}$ at the top of the neural network by means of probabilistic information transfer. This entire space multi-task modeling method makes it perform better than MLP in both CVR and CTCVR tasks. However, ESMM has not considered the commonalities and differences between CTR and CVR prediction tasks at the bottom of the MLP. AITM uses the AIT and calibrator modules to improve CTCVR task with CTR task, but it cannot completely avoid the phenomenon of $p_{ctcvr} > p_{ctr}$ because the calibrator can only act as a regularizer rather than restricting it from the probability formula. In addition, this method does not explicitly or implicitly model the CVR task, which makes its low performance in the CVR task. TAML achieves state-of-the-art in all tasks, which is not only due to the improvement achieved by the Multi-Learner module learning knowledge from different views but also because of the self-distillation loss that encourages the knowledge exchange among learners. And the HTAKR module has better performance in extracting knowledge representation information for each task.

### 3.4 Online Evaluation

The proposed method and baseline MTL method were trained offline and regularly updated from April 9 to April 21. Each pre-trained model is deployed in a single cluster to real-time show advertising for mobile app promotion. For online serving, we randomly select 2% of the users as the experimental group that is recommended ads by TAML and another 2% of the users as the control group that is recommended ads by baseline MTL method. We use CVR and Effective Cost per Mille (eCPM) to evaluate the performance of the above deployed models.

The online A/B test results of consecutive 15 days show significant improvement of TAML over the baseline MTL model. Compared with the baseline MTL model, the conversion rate increases by 2.49%, and eCPM increases by 2.41%. It brings significant business revenue improvement with slight latency overload. Now, TAML has provided real-time prediction for the major traffic in our system.

### 3.5 Ablation Study

To validate the effectiveness of each module in TAML, we conduct ablation studies on the Ali-CCP dataset. The experimental results are shown in Fig. 3. From the results, we find each module plays an important role in our method, and removing each of them will lead to performance degradation. Among them, the multi-learner mechanism has the most salient contribution. This is because using multiple learners can help encode more comprehensive and less biased knowledge. The self-distillation mechanism also has a notable contribution. It may be because using knowledge distillation to fuse the knowledge extracted from multiple learners is better than simply increasing model capacity. The HTAKR module also has some contributions. This shows that disentangling different types of knowledge is more suitable for multi-task learning.

### 3.6 Gating Weight Analysis

Here, we investigate the gating weights of the three types of experts on Ali-CCP, as shown in Fig. 4. We find different tasks have diverse gating weights. In the CTR task, general experts are less important than task-specific experts, while on the contrary in the CVR task. This is because the CTR task can provide rich supervision signals to assist CVR task in distinguishing a large number of negative impression samples so that the CTR task is relatively more helpful to the CVR task. The learner-specific experts play important roles in both CTR and CVR tasks. It shows the importance of fine-grained learner-specific knowledge in CTR and CVR prediction.

### 4 CONCLUSION

In this paper, we propose a task adaptive multi-learner (TAML) framework for joint CTR and CVR estimation. TAML uses a hierarchical task adaptive knowledge representation module to disentangle knowledge in different granularities, and compose them adaptively as the input of multiple learners. The learners of each task generate final predictions collaboratively, and they exchange their hidden knowledge via self-distillation. Experiments on both public and proprietary datasets show that TAML outperforms the previous state-of-the-art models on three important tasks CTR, CVR, and CTCVR. Meanwhile, online experiments on Huawei’s advertising platform show that our model achieves notable online improvements, and it is deployed to serve the personalized Ad recommendation service for App promotion.
REFERENCES


