Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space

Pengcheng Li*
lipc@lamda.nju.edu.cn
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, China

Runze Li*
runze.lrz@alibaba-inc.com
Alibaba Group
Hangzhou, China

Qing Da
daqing.dq@alibaba-inc.com
Alibaba Group
Hangzhou, China

An-Xiang Zeng
renzhong@taobao.com
Alibaba Group
Hangzhou, China

Lijun Zhang
zhanglj@lamda.nju.edu.cn
National Key Lab for Novel Software Technology, Nanjing University
Nanjing, China

ABSTRACT

Traditional Learning to Rank (LTR) models in E-commerce are usually trained on logged data from a single domain. However, data may come from multiple domains, such as hundreds of countries in international E-commerce platforms. Learning a single ranking function obscures domain differences, while learning multiple functions for each domain may also be inferior due to ignoring the correlations between domains. It can be formulated as a multi-task learning problem where multiple tasks share the same feature and label space. To solve the above problem, which we name *Multi-Scenario Learning to Rank*, we propose the **H**ybrid of *implicit* and *explicit* **M**ixture-**o**f-**E**xperts (HMoE) approach. Our proposed solution takes advantage of Multi-task Mixture-of-Experts to *implicitly* identify distinctions and commonalities between tasks in the feature space, and improves the performance with a stacked model learning task relationships in the label space *explicitly*. Furthermore, to enhance the flexibility, we propose an end-to-end optimization method with a task-constrained back-propagation strategy. We empirically verify that the optimization method is more effective than two-stage optimization required by the stacked approach. Experiments on real-world industrial datasets demonstrate that HMoE significantly outperforms the popular multi-task learning methods. HMoE is in-use in the search system of AliExpress and achieved 1.92% revenue gain in the period of one-week online A/B testing. We also release a sampled version of our dataset to facilitate future research.

*Both authors contributed equally to this research.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412713

CCS CONCEPTS

• **Computing methodologies** → **Multi-task learning**; **Learning to rank**; **Neural networks**.

KEYWORDS

multi-task learning, learning to rank, neural networks, e-commerce

ACM Reference Format:
Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412713

1 INTRODUCTION

Since E-shopping becomes increasingly widespread, E-commerce search has attracted more attention recently and has been studied extensively [1, 2, 7, 9, 21]. Learning to Rank (LTR) involves applying machine learning algorithms in optimizing the rank strategy, and is the fundamental technique to facilitate better services for product searching. In the context of E-commerce search, the feedback of users, such as clicking and purchasing, is treated as an implicit relevance signal instead of obtaining relevance labels by human judgement in traditional web search.

Although there has been much progress made in research on Learning to Rank, the majority of them are based on the hypothesis that data are from a single domain. However, this hypothesis is usually violated in a practical situation. For example, in international E-commerce platforms like Amazon and AliExpress, behaviours of users from different countries are diverse due to different languages, races, and regions. User behaviours can be seen as instances drawn from multiple domains with domain-specific distributions. Training a single LTR model for the whole data might be confused by the distinctions between tasks. While training multiple models for each task separately may also be inferior, especially for tasks whose training data is scarce. Many existing Multi-Task Learning (MTL) methods [11, 14, 18] can solve the problem by learning commonalities and differences between tasks. In this paper, we consider a specific MTL setting where tasks share the same feature and label space. We name the problem *Multi-Scenario Learning to Rank* and a task is named a *scenario*. Sharing a common label space implies that the objective is identical for all the scenarios, e.g., maximizing Click-Through Rate (CTR) for all the countries. Most prior MTL models only learn task relationships in the feature space, while none of them sufficiently exploits the characteristics that tasks share the same label space.

In this paper, we demonstrate that the performance of one scenario can be enhanced by the prediction of the other scenarios, which implies the effectiveness of utilizing the characteristics of sharing a common label space between scenarios. We propose the **H**ybrid of implicit and explicit **M**ixture-**o**f-**E**xperts (HMoE) model. HMoE employs Multi-task Mixture-of-Experts (MMoE) [16] to identify differences and similarities between tasks in the feature space implicitly, and improve the performance with a stacked approach [3] learning task relationships in the label space explicitly. The stacked technology requires two-stage optimization which prohibits the model from training on real-time data continuously to track dynamic user intention in a timely manner [6]. To this end, we propose an end-to-end optimization method with a task-constrained back-propagation strategy to preserve the domain-specific knowledge of scenario predictions. We summarize our main contributions as follows:

- We empirically verify the effectiveness of leveraging the property of sharing a common label space between scenarios. With the above benefit, we propose HMoE based on the MMoE with a stacked model and adopt an end-to-end optimization method to train the model continuously on fresh data.
- The task-constrained back-propagation strategy makes the relationships of scenarios in the label space explainable. We empirically demonstrate that the strategy is clearly essential to obtain promotion.
- We conduct extensive experiments to demonstrate that HMoE significantly improves the performance on industrial datasets. Our proposed model serves the ranking system of AliExpress [1] search engine and increases revenue by 1.92% in online A/B testing. We also release a sampled version of our dataset to facilitate future research.

2 RELATED WORK

2.1 Learning to Rank

Learning to Rank (LTR) [4, 15] is a fundamental problem for recommender, search, and advertising. Ranking strategies greatly affect user experience and advertising revenue. Traditional LTR models are usually trained on substantial data based on i.i.d. hypothesis in a batch mode. In recent years, there is a growing number of research on studying LTR with deep neural networks. Haldar et al. [9] applied Deep Neural Network (DNN) in breaking out of the plateau of Gradient Boosted Decision Tree (GBDT) for Airbnb search. Ai et al. [1] used the inherent feature distributions of the top results to learn a deep list-wise context model that can fine-tune the initial ranked list.

Studies above can achieve good performance on average but may be sub-optimal when data come from multiple domains with domain-specific distributions. Bai et al. [2] firstly proposed the multi-task learning to rank problem for web search, while they did not exploit task relationships in the label space. Ni et al. [19] learned universal user representations across multiple search and recommendation tasks for more effective personalization. Feng et al. [7] improved the overall performance of ranking strategies in search, recommendation, and advertising by multi-agent reinforcement learning. Chapelle et al. [6] claimed that transfer LTR and online LTR are the future research directions. Ktena et al. [13] addressed delayed feedback for continuous training with neural networks in CTR prediction.

2.2 Multi-Task Learning

Multi-Task Learning (MTL) [5] is a powerful technology when there are multiple related tasks. It can learn commonalities and differences across different tasks [5, 20]. Caruana [5] proposed a widely used multi-task learning model, which has a shared-bottom model structure. This structure may suffer from optimization conflicts because of the difference between tasks. Lots of recent approaches add different types of constraints on task-specific parameters. For example, Liu et al. [14] alleviated the shared and private latent feature spaces from interfering by adding the adversarial loss for two tasks. Misra et al. [18] learned an unique combination of task-specific hidden-layers for each task which is called cross-stitch network.

Several approaches try to capture the relationship between different tasks. Kendall et al. [11] took an orthogonal approach by considering the uncertainty of each task and adjust each weight in the cost function by deriving a multi-task loss function based on maximizing the Gaussian likelihood with task-dependent uncertainty. Ma et al. [16] proposes Multi-gate Mixture-of-Experts (MMoE) to model task relationships from data. However, the relationship between different scenarios is implicitly learned in experts and gating functions and none of these methods explicitly exploits the property of the same label space.

3 THE PROPOSED APPROACH

In this section, we introduce the Multi-Scenario Learning to Rank problem and present our proposed model, named Hybrid of implicit and explicit Mixture-of-Experts (HMoE), in detail. Our model consists of two parts: the first part aims to capture the similarities and differences between scenarios with Multi-task Mixture-of-Experts in the feature space; the second part aims to extract scenario correlations in the label space explicitly with a stacked model. We further design an elegant end-to-end optimization approach to enhance flexibility and facilitate the model to be trained online.

3.1 Problem Formulation

We formulate *Multi-Scenario Learning to Rank* as a problem that given T learning to rank tasks $\{Q_t\}_{t=1}^{T}$ with a common feature space $\mathcal{X}$ and label space $\mathcal{Y}$ and all the tasks or a subset of them are related. Each task Q_t is a *scenario*. Suppose that t-th scenario has N_t labeled training data

$$C_t = \left\{ (x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \cdots, (x_{tN_t}, y_{tN_t}) \right\}, \tag{1}$$

[1] https://www.aliexpress.com/

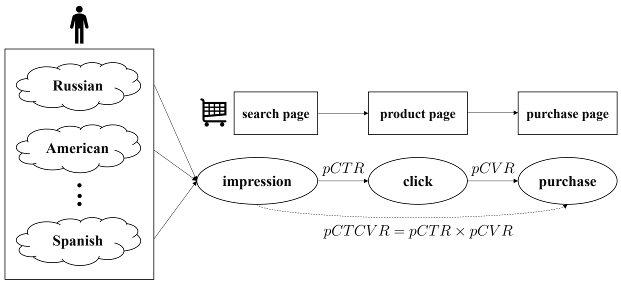

Figure 1: Illustration of a search session in our e-commerce platform. Users from different countries can be viewed as instances sampled from different scenarios. Generally, we assume that user behaviour mainly consists of two stages. An user firstly clicks a product from the search page, and then decides to purchase the product or not. We focus on how to maximize post-view click-through&conversion rate (CTCVR).

that are drawn from a domain-specific distribution P_t over $\mathcal{X} \times \mathcal{Y}$. P_t can be varying for different scenarios. The goal is to construct an unified ranking function h: $\mathcal{X} \rightarrow \mathcal{Y}$ that can accurately predict for each scenario simultaneously. In our setting, training sets are gathered from different countries, e.g., C_1 is from Russia and C_2 is from America. We denote the loss function for t-th scenario as $\ell(y_t, h_t)$. The empirical risk can be formulated as

$$\mathcal{R}(h) = \sum_{t=1}^{T} \sum_{i=1}^{N_t} \ell\left(y_{ti}, h\left(x_{ti}\right)\right). \quad (2)$$

As shown in Figure 1, for E-commerce search ranking, a search session mainly consists of two stages. The first one is an user searching by a query and selecting a product to click. The second one is deciding whether to purchase the product after viewing the detailed product page. There is a sequential dependence between purchase and click. That is, a purchased product must have been clicked. Our goal is to maximize post-view click-through&conversion rate (CTCVR), i.e. $pCTCVR = Pr(purchase, click|impression)$. We decompose the objective of predicting $pCTCVR$ into two sub-objectives as mentioned above: (1) the **click model** predicting post-view click-through rate (CTR), i.e. $pCTR = Pr(click|impression)$ and (2) the **purchase model** predicting post-click conversion rate, i.e. $pCVR = Pr(purchase|click, impression)$. Therefore the conditional probability can be written as

$$pCTCVR = \underbrace{pCTR}_{\text{click model}} \times \underbrace{pCVR}_{\text{purchase model}}. \quad (3)$$

The ranking list is ordered by the product of outputs of two models.

3.2 Implicit Multi-task Mixture-of-Experts Model

In order to improve the performance of each scenario, we employ Implicit Multi-task Mixture-of-Experts (IMMoE) [16] for the ability to model tradeoffs between scenario-specific objectives and inter-scenario relationships. In this section, we introduce IMMoE briefly. The model structure is shown in Figure 2 (a).

The original Mixture-of-Experts (MoE) model [10] is an ensemble structure which consists of multiple expert networks $E_i(x) = f_e(x), i = 1, 2, \cdots K$ and a gate network $G(x) = f_g(x)$. The gate network assemble the results from experts, which is a weighted sum of the outputs of all the experts:

$$M(x) = \sum_{i=1}^{K} G_i(x) E_i(x), \quad (4)$$

where $G_i(x)$ is the i-th element of $G(x)$ which represents the probability of expert $E_i(x)$ and $\sum_{i=1}^{K} G_i(x) = 1$.

IMMoE is a shared-bottom multi-task model that adapts MoE as the shared-bottom network to learn task relationships and includes a successive tower network for each task respectively to learn task-specific knowledge. We denote the prediction of t-th task as

$$S_i(x) = f_s(M(x)). \quad (5)$$

Instead of all tasks using the same set of parameters on shared-bottom layers, IMMoE takes advantage of MoE that alleviates optimization conflicts caused by task differences. The outputs of experts have no domain-specific meaning. That is, IMMoE implicitly learns task relationships in the feature space, and we say it is "implicit". We use IMMoE as our base model.

3.3 Hybrid of Implicit and Explicit Mixture-of-Experts

IMMoE implicitly identifies the relationships between scenarios with multiple experts in the feature space. However, based on the property of scenarios sharing a common label space and the observation that the best performance for a scenario is achieved by the model trained on another scenario in several cases (as shown in the experimental section), we propose a novel hybrid model by learning scenario relationships in the label space. We present the model structure in Figure 2 (b) and (c).

Considering two related scenarios Q_i and Q_j, a low-density area in the distribution P_i is corresponding to a high-density area in the distribution P_j. The prediction of IMMoE $S_i(x)$ may make a mistake with a high probability for an instance x sampled from the low-density area in P_i. But if instance x is treated as an instance sampled from the high-density area in P_j, it may be predicted accurately. In this situation, the performance of scenario Q_i can be promoted by the aid of prediction of scenario Q_j.

The promotion is connected with the confidence of prediction for x and the strength of scenarios' correlation. Therefore, we utilize a stacked approach [3] to learn how the prediction of one scenario can be rectified by predictions of others. The stacked model assembles predictions of different scenarios $S_i(x)$ with a scenario gate network $W(x) = f_w(x)$. We denote the output of this model as

$$H(x) = \sum_{i=1}^{T} W_i(x) S_i(x), \quad (6)$$

where $W_i(x)$ is the i-th element of $W(x)$ which represents the importance of i-th scenario's prediction $S_i(x)$ and $\sum_{i=1}^{T} W_i(x) = 1$. Note that the meaning of $W_i(x)$ is related to which scenario x belongs to, e.g., $W_i(x_j)$ indicates the weight of i-th scenario for j-th scenario.

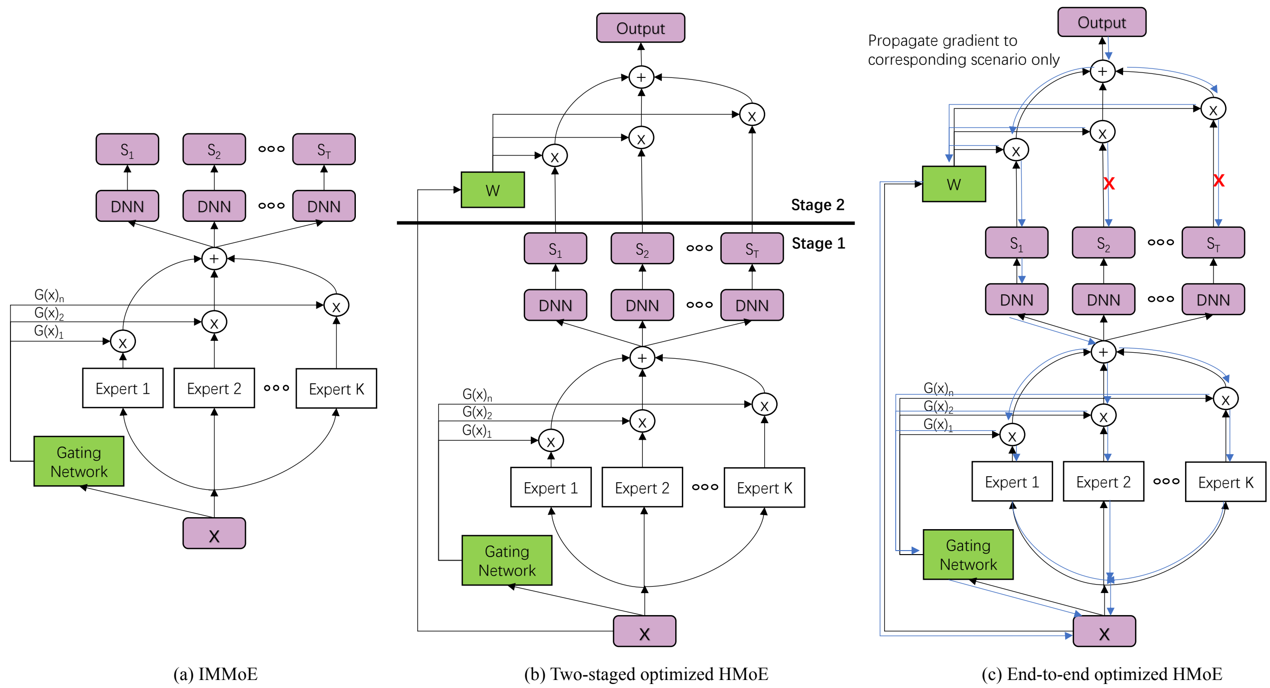

Figure 2: The diagram of (a) IMMoE. (b) HMoE with two-stage optimization. (c) Jointly optimized HMoE. The blue lines illustrate the back-propagation of the gradients on the data of *the 1st scenario*. The red crosses represent prohibiting the gradients from being back-propagated.

In the standard stacked approach, HMoE is trained by two stages, as shown in Figure 2 (b). The first stage is the training process of IMMoE. In the second stage, we only update the parameters in the scenario gate network $W(x)$ to learn relationships between scenarios in the label space. This method makes the model optimized in the second stage similar to the MoE structures, i.e. each scenario network $S_i(x)$ plays a role like an expert network $E_i(x)$ and the role of the gate network $W(x)$ is similar to $G(x)$. But the difference is that we can obtain relatedness between scenarios in the label space from $W(x)$ explicitly.

We name this model Hybrid of implicit and explicit Mixture-of-Experts (HMoE) where "implicit" represents implicit relationships in the feature space from IMMoE and "explicit" represents explicit relatedness in the label space from the stacked model.

3.4 End-to-end optimization

We can obtain a better prediction quality by the stacked model. Nevertheless, the way of stacked optimization has several drawbacks. Firstly, to avoid overfitting, the traditional stacked technique requires two disjoint training set for two optimization stages. It impedes sufficiently training of the whole model, especially for those scenarios with a small amount of training data. Secondly, the flexibility of the model is hurt since it desires two disjoint optimization processes and can not be trained online. In the context of E-commerce search or recommendation, users' intentions could vary dramatically along with time, especially for those time-sensitive queries. It is essential to train the model with users' real-time feedback data continuously to trace users' dynamic interests.

If we directly combine the two-stage optimization into a joint optimization approach, the explicit domain-specific knowledge of each scenario's prediction $S_i(x)$ will disappear, which we name *scenario awareness missing* problem. We argue that reserving the explicit domain-specific knowledge of each scenario's prediction has a positive effect on the performance, which is validated in the experimental section.

To solve this problem, we propose an elegant and flexible end-to-end optimization method. We apply a domain-specific constraint to the back-propagation of gradients. For example, if an instance (x_{ti}, y_{ti}) belongs to a scenario Q_t, we prohibit the gradients from being back-propagated to the parameters in scenario networks $S_j(x)$, where $j = 1, 2, \cdots, T$ and $j \neq t$. The domain-specific knowledge of scenario prediction $S_t(x)$ is preserved through the scenario constrained gradients. The back-propagation of gradients is shown in Figure 2 (c). The prediction for scenario Q_t can be formulated as

$$H_t(x) = W_t(x)S_t(x) + \sum_{j=1, j\neq t}^{T} W_j(x) \underbrace{S_j(x)}_{\text{stop gradient}} . \quad (7)$$

4 EXPERIMENTS

In this section, we firstly verify the effectiveness of leveraging the characteristics of sharing a common label space for all the scenarios. Then we conduct experiments to demonstrate the improvement of

HMoE on two datasets: the first dataset is collected from real-world traffic log of the search system in AliExpress; the second one, a public dataset called Ali-CCP [2] [17] is gathered from the recommender system in Taobao which is the largest online shopping platform in China.

4.1 Experimental Setup

Datesets We collect **Industrial AE** dataset from traffic logs of AliExpress search system where we regard each country as a scenario. We split the first 90% data in the time sequence to be training set while the rest 10% to be test data. We further release a randomly sampled version of the whole dataset as the **Public AE** dataset [3] and split the training and test dataset in the same manner. The second dataset, **Ali-CCP**, is provided for the purpose of CTR and CVR prediction tasks. We convert an LTR task into a CTR prediction on this dataset. According to the authors' description, the dataset contains three types of scenarios which correspond to three kinds of the context feature value. Table 1 summarizes the statistics of the three datasets.

	Public AE	Industrial AE	Ali-CCP
#user	3.6M	17.9M	0.4M
#product	26.5M	0.7B	4.3M
#pv	15.4M	0.7B	-
#impression	2.3B	9.9B	84M
#click	5.8M	0.3B	3.4M
#purchase	0.1M	1.2M	18K

Table 1: Statistics of experimental datasets. #pv, #product and #user represent the number of product ranking lists, products and users respectively. #impression, #click and #purchase are the number of users' browsing, click and purchasing separately.

In AE dataset, we collect data from 5 representative countries, i.e. Russia (RU), Spain (ES), French (FR), Netherlands (NL), and America (US), each of which refers to a scenario. Table 2 shows the statistics of each scenario in Public AE. The differences in user behaviours between these 5 countries are obvious. Users from RU are more likely to click on ranking lists but do not easily purchase after a click. Users from NL are most likely to purchase after a click.

Compared models We conduct experiments with several compared methods for a LTR task: maximizing CTCVR. These methods can be divided into three groups:

- **Single-task learning method**: (1) *BaseDNN* is a DNN, layers of which is set by 128-64-32. (2) *Scenario-DNN* consists of multiple independent DNNs, each of which is trained for one scenario. (3) *Fine-tune-DNN* is a Scenario-DNN and each scenario's model is initialized with BaseDNN pre-trained on all the scenarios. (4) *DeepFM* [8] is a factorization-machine based neural network.
- **Multi-task learning method**: (5) *Cross-stitch* [18] uses linear cross-stitch units to learn an optimal combination of shared and task-specific representations. (6) *UncertWeight* [11] weighs multiple loss functions by considering the homoscedastic uncertainty of each task. (7) *AdvLoss* [14] alleviates the shared and private latent feature spaces from interfering with each other tasks by adding the adversarial loss. (8) *IMMoE* is the base model we use.
- **Variant of HMoE**: (9) *Two-stage-HMoE* is HMoE optimized in two stages. (10) *HDNN* replaces the shared-bottom network of HMoE from IMMoE to BaseDNN. (11) *NoStopGrad-HMoE* is HMoE without the constraint of stop gradient.

[2]https://tianchi.aliyun.com/datalab/dataSet.html?dataId=408
[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=74690

	RU	ES	FR	NL	US
#product	16.7M	8.7M	7.4M	6M	8M
#pv	8.7M	2M	1.7M	1.2M	1.8M
#impression	1.3B	31.6M	27.4M	17.7M	27.4M
#click	3.6M	841K.	535K	382K	450K
#purchase	61.8K	19.1K	14.4K	13.8K	10.9K
CTR	2.78%	2.66%	2.01%	2.16%	1.64%
CVR	1.71%	2.27%	2.42%	3.61%	2.42%
CTCVR	0.48‰	0.60‰	0.54‰	0.78‰	0.40‰

Table 2: Statistics of each scenario in Public AE, where CTR = #click/#impression, CVR = #purchase/#click and CTCVR = #purchase/#impression.

Training details We multiply the output of click models by the output of purchase models as the final ranking score. To make the product of the two outputs represent $pCTCVR$, the two models are required to estimate the actual CTR/CVR values accurately. Therefore we use pointwise cross entropy loss to train the models. The two models use separate parameters and use the same model structure. We apply 5 experts in HMoE and each expert is a 1-layer (128) network. The two gating networks are both fully connected networks with 1 hidden layer (64) and the outputs are activated by softmax function. $S_i(x)$ is a 2-layer (64-32) network. The activation function of expert networks and scenario network is ReLU. The base model of Cross-stitch, UncertWeight, and AdvLoss is BaseDNN. The optimizer used in our experiments is Adam [12] with default parameters.

Metrics Area under the ROC curve (AUC) and Normalized Discounted Cumulative Gain (NDCG) are applied extensively in CTR prediction and LTR field respectively. Another metric we used is called Group AUC (GAUC), which is the average of AUC on each product ranking list: $GAUC = \frac{1}{N}\sum_{i=1}^{N} AUC_i$, where AUC_i is AUC on i-th product ranking list. As claimed in [22], the first stage, predicting $pCTR$, is a ranking problem, and the second stage, predicting $pCVR$, is a binary classification problem. Therefore we use GAUC and NDCG to evaluate performance of click model and the final ranking lists ordered by $pCTCVR$, while AUC is used to evaluate the performance of purchase models on AE dataset. We only apply AUC on Ali-CCP dataset for the reason that there is no information about ranking lists. We use CTR-X, CVR-X, CTCVR-X to refer to the metric X of click model, purchase model and two joint models respectively.

4.2 Verification of Effectiveness

We conduct an experiment on Industrial AE dataset to verify the effectiveness of the intuition that the performance of MTL model

Group of Model	Model	CTR-GAUC	CTR-NDCG@2	CTR-NDCG@5	CTR-NDCG@10	CTR-NDCG@17
Single-task	BaseDNN	0.6396	0.2604	0.3562	0.4331	0.4396
	Scenario-DNN	0.6559	0.2857	0.3807	0.4542	0.4595
	Fine-tune-DNN	0.6531	0.2785	0.3748	0.4495	0.4545
	DeepFM	0.6548	0.2803	0.3798	0.4517	0.4576
Multi-task	Cross-stitch	0.6610	0.2952	0.3898	0.4605	0.4593
	UncertWeight	0.6570	0.2872	0.3823	0.4557	0.4608
	AdvLoss	0.6538	0.2879	0.3808	0.4542	0.4604
	IMMoE	0.6608	0.2945	0.3879	0.4594	0.4648
Variant	Two-stage-HMoE	0.6617	0.2933	0.3906	0.4611	0.4634
	NoStopGrad-HMoE	0.6596	0.2954	0.3885	0.4611	0.4644
	HDNN	0.6524	0.2871	0.3816	0.4552	0.4601
Ours	HMoE	**0.6650**	**0.2982**	**0.3923**	**0.4632**	**0.4680**

Table 3: Comparison results of different click models on Public AE Dataset.

can be improved by learning scenario relationships in the label space in our setting. We train simple DNN click models for each scenario and overall dataset with the same structure and the same amount of data. Overall dataset is sampled to align the amount of data with each scenario. The results on the test dataset of each scenario are shown in Table 4 and 5.

Train	Test: ALL	RU	ES	FR	NL	US
ALL	0.708	0.7123	0.7073	0.7095	0.7098	0.6998
RU	0.7249	**0.7288**	0.7236	**0.7264**	0.7245	**0.7174**
ES	**0.7251**	0.7274	**0.7257**	0.7246	**0.7259**	0.7162
FR	0.7225	0.7252	0.7233	0.726	0.7244	0.7146
NL	0.7186	0.7214	0.7195	0.7226	0.7229	0.712
US	0.7196	0.723	0.7171	0.7221	0.7222	0.7139

Table 4: Heatmap of CTR-AUC. The deeper the color, the higher the value. ALL represents sampled overall dataset.

Train	Test: ALL	RU	ES	FR	NL	US
ALL	0.4391	0.4489	0.4438	0.4362	0.4409	0.4339
RU	**0.4587**	**0.4633**	0.4502	**0.4537**	0.4493	0.4489
ES	0.4542	0.4575	0.4513	0.4523	0.4487	**0.4522**
FR	0.4508	0.4582	0.4506	0.4457	0.4499	0.4473
NL	0.4558	0.4576	0.4503	0.4469	0.4521	0.4517
US	0.4486	0.4604	**0.4553**	0.4476	**0.4517**	0.4472

Table 5: Heatmap of CTR-NDCG@10.

CTR-AUC and CTR-NDCG of the model trained on ALL dataset are both the worst for all the counties, which indicates that the performance of models ignoring scenario relationships is inferior. The other fact is that the best performance for a scenario is not always obtained by the model trained with its own data, e.g., the highest CTR-AUC and CTR-NDCG for scenario FR is achieved by the model trained with data from scenario RU. It is implied that we can improve the performance of one scenario through exploiting the prediction of the other related scenarios.

4.3 Experimental Results

Promotion on each scenario We examine the promotion of HMoE compared with IMMoE by learning scenario relationships in the label space in each scenario on Industrial AE dataset. The comparison results of click models are shown in Table 6.

Scenario	IMMoE	HMoE	Relative Improved
RU	0.6610	**0.6658**	0.73%
ES	0.6566	**0.6626**	0.91%
FR	0.6585	**0.6638**	0.80%
NL	0.6542	**0.6591**	0.75%
US	0.6518	**0.6570**	0.80%
ALL	0.6566	**0.6619**	0.81%

Table 6: CTR-GAUC of IMMoE and HMoE on each scenario.

HMoE achieves better performance in all the scenarios in different degrees. The gain of HMoE obtained in scenario RU is the smallest, corresponding to the fact that the model trained with RU's data achieves the best performance in other scenarios.

Results from comparison models on three datasets The comparison results of click models on Public AE are shown in Table 3. The results of click models, purchase models and the two joint models on Industrial AE are shown in Table 7, 9 and 8 respectively. Table 10 shows the results of click models on Ali-CCP. Our model, HMoE, beats all the compared models and shows the best performance on all the datasets.

The comparison results on these datasets consistently demonstrate several conclusions:

- **The impact of domain differences:** as expected, training multiple models for each scenario and MTL models are both achieve better performance than training a single model. While MTL models are not always superior to single-task models. One possible reason is that MTL models fail to understand the distinctions between scenarios and involve negative knowledge transfer.
- **The effectiveness of learning scenario correlations in the label space:** both HMoE and Two-stage-HMoE promotes the prediction quality of IMMoE. Compared with Two-stage-HMoE, HMoE is more flexible and effective by reason

Group of Model	Model	CTR-GAUC	CTR-NDCG@2	CTR-NDCG@5	CTR-NDCG@10	CTR-NDCG@17
Single-task	BaseDNN	0.6446	0.2827	0.3813	0.4486	0.4495
	Scenario-DNN	0.6537	0.2866	0.3852	0.4544	0.4578
	Fine-tune-DNN	0.6495	0.2852	0.3826	0.4491	0.4545
	DeepFM	0.6531	0.2874	0.3831	0.4527	0.4564
Multi-task	Cross-stitch	0.6604	0.2955	0.3864	0.4607	0.4593
	UncertWeight	0.6528	0.2830	0.3806	0.4509	0.4514
	AdvLoss	0.6583	0.2896	0.3882	0.4574	0.4596
	IMMoE	0.6566	0.2874	0.3860	0.4563	0.4559
Variant	Two-stage-HMoE	0.6609	0.2942	0.3893	0.4618	**0.4630**
	NoStopGrad-HMoE	0.6594	0.2902	0.3889	0.4564	0.4560
	HDNN	0.6543	0.2908	0.3851	0.4565	0.4565
Ours	HMoE	**0.6619**	**0.2960**	**0.3947**	**0.4627**	0.4613

Table 7: Comparison results of different click models on Industrial AE Dataset.

Group of Model	Model	CTCVR-GAUC	CTCVR-NDCG@2	CTCVR-NDCG@5	CTCVR-NDCG@10	CTCVR-NDCG@17
Single-task	BaseDNN	0.6532	0.3519	0.4410	0.4723	0.5135
	Scenario-DNN	0.6628	0.3609	0.4477	0.4790	0.5228
	Fine-tune-DNN	0.6605	0.3585	0.4452	0.4766	0.5205
	DeepFM	0.6607	0.3590	0.4456	0.4770	0.5208
Multi-task	Cross-stitch	0.6704	0.3655	0.4518	0.4837	0.5273
	UncertWeight	0.6596	0.3577	0.4468	0.4781	0.5196
	AdvLoss	0.6644	0.3627	0.4492	0.4808	0.5245
	IMMoE	0.6657	0.3642	0.4505	0.4819	0.5256
Variant	Two-stage-HMoE	0.6701	0.3648	0.4510	0.4831	0.5269
	NoStopGrad-HMoE	0.6709	0.3657	0.4522	0.4838	0.5276
	HDNN	0.6610	0.3592	0.4459	0.4774	0.5213
Ours	HMoE	**0.6731**	**0.3680**	**0.4543**	**0.4857**	**0.5292**

Table 8: Comparison results of different click models and purchase models on Industrial AE Dataset.

Group of Model	Model	CVR-AUC
Single-task	BaseDNN	0.8287
	Scenario-DNN	0.8436
	Fine-tune-DNN	0.8359
	DeepFM	0.8335
Multi-task	Cross-stitch	0.8499
	UncertWeight	0.8371
	AdvLoss	0.8422
	IMMoE	0.8441
Variant	Two-stage-HMoE	0.8484
	NoStopGrad-HMoE	0.8510
	HDNN	0.8348
Ours	HMoE	**0.8613**

Table 9: Comparison of different purchase models in terms of CVR-AUC on Industrial AE dataset.

Group of Model	Model	CTR-AUC
Single-task	BaseDNN	0.5940
	Scenario-DNN	0.6072
	Fine-tune-DNN	0.5968
	DeepFM	0.6032
Multi-task	Cross-stitch	0.6161
	UncertWeight	0.6082
	AdvLoss	0.6059
	IMMoE	0.6118
Variant	Two-stage-HMoE	0.6073
	NoStopGrad-HMoE	0.6144
	HDNN	0.5971
Ours	HMoE	**0.6196**

Table 10: Comparison of different models in terms of CTR-AUC on Ali-CCP dataset.

that HMoE can be trained on whole data sufficiently in its entirety.

- **The effectiveness of reserving the domain-specific knowledge of scenario predictions:** we compare HMoE with HDNN and NoStopGrad-HMoE and the experimental results prove the effectiveness of domain-specific knowledge of scenario predictions preserved by the task-constrained back-propagation strategy.

Scenario relationships in the label space We validate whether HMoE identities scenario relationships in the label space by visualizing the output of gate network $W(x)$.

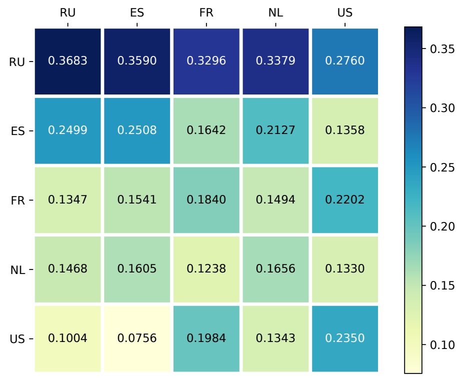

Figure 3: Visualization of scenario correlations in the label space in click model.

Figure 3 displays the heatmap of prediction of $W(x)$ in click model. The element in i-th row and j-th column represents the average value of $W_i(x_j)$ on test data of j-th scenario. Outputs of gate networks between FR and US are largest for each other, corresponding to the fact that CTR of FR and US are lowest. The weights between RU, ES and NL are similar, which are consistent with the results in the previous section.

Online deployment and A/B testing HMoE has been deployed in the search system of AliExpress. Limited by computation cost, we divide our data into 5 scenarios, i.e. the 4 countries with the largest amount of data and the union of the rest of countries. We conduct one-week online A/B testing with the objective of maximizing CTCVR. Table 11 shows the relative improvement by HMoE. This is a significant promotion for industrial applications where 0.1% gain of revenue is considerable.

Metric	CTR	CVR	CTCVR	GMV
Relative Improved	0.12%	1.27%	1.40%	1.92%

Table 11: Improvement of HMoE compared with BaseDNN in online A/B testing. GMV refers to Gross Merchandise Volume.

5 CONCLUSION

In this paper, we propose the Multi-Scenario Learning to Rank problem and verify the effectiveness of leveraging the property of sharing a common label space between scenarios. We propose HMoE to solve the problem which improves the performance of MMoE with a stacked model exploiting scenario relationships in the label space. We further design an end-to-end optimization method to enable the model to be trained continuously. We conduct extensive experiments to evaluate our method. The empirical results demonstrate that HMoE significantly improves the performance on both public and real-world industrial datasets. We also release a sampled version of our dataset to facilitate future research.

REFERENCES

[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning a Deep Listwise Context Model for Ranking Refinement. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. MI, USA, 135–144.

[2] Jing Bai, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, Gordon Sun, Belle L. Tseng, Zhaohui Zheng, and Yi Chang. 2009. Multi-task learning for learning to rank in web search. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. Hong Kong, China, 1549–1552.

[3] Leo Breiman. 1996. Stacked Regressions. *Machine Learning* 24, 1 (1996), 49–64.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, Vol. 227. Oregon, USA, 129–136.

[5] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[6] Olivier Chapelle, Yi Chang, and Tie-Yan Liu. 2010. Future directions in learning to rank. In *Proceedings of the Yahoo! Learning to Rank Challenge*. Haifa, Israel, 91–100.

[7] Jun Feng, Heng Li, Minlie Huang, Shichen Liu, Wenwu Ou, Zhirong Wang, and Xiaoyan Zhu. 2018. Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. Lyon, France, 1939–1948.

[8] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. Melbourne, Australia, 1725–1731.

[9] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C. Turnbull, Brendan M. Collins, and Thomas Legrand. 2019. Applying Deep Learning to Airbnb Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. AK, USA, 1927–1935.

[10] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

[11] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weight losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. UT, USA, 7482–7491.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Conference Track Proceedings of 3rd International Conference on Learning Representations*. CA, USA.

[13] Sofia Ira Ktena, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszár, Steven Yoo, and Wenzhe Shi. 2019. Addressing delayed feedback for continuous training with neural networks in CTR prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. Copenhagen, Denmark, 187–195.

[14] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada, 1–10.

[15] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.

[16] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, UK, 1930–1939.

[17] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. 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*. MI, USA, 1137–1140.

[18] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. NV, USA, 3994–4003.

[19] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. 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*. London, UK, 596–605.

[20] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR* abs/1706.05098 (2017).

[21] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2017. On Application of Learning to Rank for E-Commerce Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Tokyo, Japan, 475–484.

[22] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. MI, USA, 365–374.