Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction

Hong Wen*
Alibaba Group
Hangzhou, Zhejiang, China 311121
qinggan.wh@alibaba-inc.com

Jing Zhang*
The University of Sydney
Darlington NSW 2008, Australia
jing.zhang1@sydney.edu.au

Yuan Wang
Alibaba Group
Hangzhou, Zhejiang, China 311121
wy175696@alibaba-inc.com

Fuyu Lv
Alibaba Group
Hangzhou, Zhejiang, China 311121
fuyu.lfy@alibaba-inc.com

Wentian Bao
Alibaba Group
Hangzhou, Zhejiang, China 311121
wentian.bwt@alibaba-inc.com

Quan Lin, Keping Yang
Alibaba Group
Hangzhou, Zhejiang, China 311121
{tieyi.lq,shaoyao}@taobao.com

ABSTRACT

Recommender system, as an essential part of modern e-commerce, consists of two fundamental modules, namely Click-Through Rate (CTR) and Conversion Rate (CVR) prediction. While CVR has a direct impact on the purchasing volume, its prediction is well-known challenging due to the Sample Selection Bias (SSB) and Data Sparsity (DS) issues. Although existing methods, typically built on the user sequential behavior path "impression→click→purchase", is effective for dealing with SSB issue, they still struggle to address the DS issue due to rare purchase training samples. Observing that users always take several purchase-related actions after clicking, we propose a novel idea of post-click behavior decomposition. Specifically, disjoint purchase-related Deterministic Action (DAction) and Other Action (OAction) are inserted between click and purchase in parallel, forming a novel user sequential behavior graph "impression→click→D(O)Action→purchase". Defining model on this graph enables to leverage all the impression samples over the entire space and extra abundant supervised signals from D(O)Action, which will effectively address the SSB and DS issues together. To this end, we devise a novel deep recommendation model named Elaborated Entire Space Supervised Multi-task Model (ESM^2). According to the conditional probability rule defined on the graph, it employs multi-task learning to predict some decomposed sub-targets in parallel and compose them sequentially to formulate the final CVR. Extensive experiments on both offline and online environments demonstrate the superiority of ESM^2 over state-of-the-art models. The source code and dataset will be released.

CCS CONCEPTS

• **Computer systems organization** → **Neural networks**; • **Information systems** → **Recommender systems**.

*Both authors contributed equally to this paper.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401443

KEYWORDS

Recommender System, Entire Space Multi-Task Learning, Post-Click Behavior Decomposition, Conversion Rate Prediction

ACM Reference Format:
Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, and Quan Lin, Keping Yang. 2020. 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 (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401443

1 INTRODUCTION

Discovering valuable products or services from massive available options on the Internet for users has become a fundamental functionality in modern online applications such as e-commerce [1, 21, 26, 28], social networking [7, 20], advertising [34, 35], *etc*. Recommender System (RS) can serve this role to provides accurate, timely, and personalized services to users [4, 17, 37]. Figure 1 shows the architecture of online recommendation in e-commerce platform. It consists of two phases, *i.e*., system recommendation and user feedback. After analyzing users' long and short-term behaviors, RS first recalls a large number of related items. Then, they are ranked and exposed to users according to several ranking metrics, *e.g*., Click-Through Rate (CTR) [34, 35], Conversion Rate (CVR) [19, 28] , *etc*. Next, when going through the recommended items, users may click on and eventually purchase the interested ones, indicating a typical user sequential behavior path "impression→click→purchase" for e-commerce transaction [19]. These feedback from users are collected by RS and used to estimate more accurate ranking metrics, which are indeed very crucial for generating high-quality recommendations in turn. In this paper, we focus on the post-click CVR estimation task.

However, two critical issues in the CVR estimation makes the task quite challenging, *i.e*., Sample Selection Bias (SSB) [31] and Data Sparsity (DS) [15]. SSB refers to the systematic difference of data distributions between training space and inference space, *i.e*., conventional CVR models are trained only on clicked samples while being used for inference on all impression samples. Intuitively, clicked samples are only a very small portion of the impression samples and are biased by user self-selection (such as clicking). Therefore, when the CVR model serving online, the SSB issue will degrade its performance. Besides, due to the relatively rare clicking

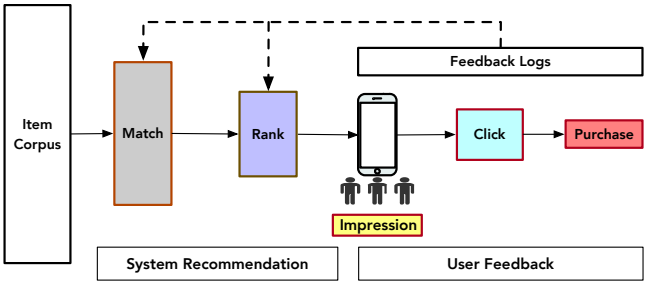

Figure 1: A diagram of online recommendation in e-commerce platform, comprising of two fundamental components, *i.e.*, system recommendation and user feedback.

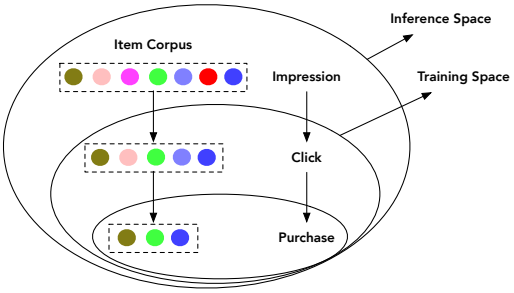

Figure 2: Illustration of sample selection bias problem in conventional CVR prediction, where training space only composes of clicked samples, while inference space is the entire space for all impression samples. And data volume gradually decreased from impression to purchase.

samples compared with impressions, the number of training samples from the sequential behavior path "click→purchase" is insufficient to fit the large parameter space of CVR task, which results in the DS problem. As illustrated in Figure 2, how to deal with the SSB and DS problems is crucial for developing an efficient industrial-level recommender system.

Several studies have been carried out to tackle these challenges [15, 19, 22, 27, 32]. For example, Ma *et al.* propose a new model named Entire Space Multi-Task Model (ESMM) [19], which defines CVR task on the user sequential behavior path "impression→click→purchase" via multi-task learning framework. It is trained with all impression samples over the entire space for two auxiliary tasks namely post-view CTR and post-view click-through conversion rate (CTCVR). Therefore, the derived CVR from CTR and CTCVR is also applicable in the same entire space when inferring online, thus addressing the SSB issue effectively. Besides, an auxiliary CTR network with rich labeled samples shares the same feature representation with the CVR network, helping to alleviate the DS issue. Although ESMM achieves better performance than conventional methods by dealing with the SSB and DS issues simultaneously, it still struggles to alleviate the DS issue due to the rare purchase training samples, *i.e.*, less than 0.1% of impression behaviors converts to purchase according to the large scale real transaction logs from our e-commerce platform.

After a detailed analysis of the logs, we observe that users always take some purchase-related actions after clicking. For example, users may add the preferred items to their shopping cart (or wish list) instead of immediately purchases due to some reasons (*i.e.*, waiting for a discount). Besides, these actions are indeed more abundant than purchase actions. Motivated by this, we propose a novel idea of post-click behavior decomposition. Specifically, disjoint purchase-related Deterministic Action (DAction) and Other Action (OAction) are inserted between click and purchase in parallel, forming a novel user sequential behavior graph "impression→click→D(O)Action→purchase", where the task relationship is explicitly defined by the conditional probability. Besides, defining model on this graph enables to leverage all impression samples over the entire space and extra abundant supervisory signals from post-click behaviors, efficiently addressing the SSB and DS issues.

In this paper, we resort to deep neural networks to embody the above idea. Specifically, we propose a novel deep neural recommendation model named Elaborated Entire Space Supervised Multi-task Model (ESM^2), which consists of three modules: 1) a shared embedding module (SEM), 2) a decomposed prediction module (DPM), and 3) a sequential composition module (SCM). First, SEM embeds a one-hot feature vector of ID features into dense representation through a linear fully connected layer. Then, these embeddings are fed into the subsequent DPM, where individual prediction network estimates the probabilities of decomposed sub-targets in parallel by employing multi-task learning on all the impression samples over the entire space. Finally, SCM composes the final CVR as well as some auxiliary probabilities sequentially according to the conditional probability rule defined on the graph. Multiple losses defined on some sub-paths of the graph are used to supervise the training of ESM^2.

The main contributions of this paper are summarized as follows:

- To the extent of our knowledge, we are the first to introduce the idea of post-click behavior decomposition to model CVR over the entire space. The explicit decomposition results in a novel user sequential behavior graph "impression→click→D(O)Action→purchase".
- We propose a novel deep neural recommendation method named ESM^2, which models CVR prediction and auxiliary tasks simultaneously in a multi-task learning framework according to the conditional probability rule defined on the user behavior graph. ESM^2 can address the SSB and DS issues efficiently by harvesting the abundant post-click action data with labels.
- Our model achieves better performance on the real-world offline dataset than representative state-of-the-art methods. We also deploy it in our online recommender system and achieve significant improvement, confirming its value in industrial applications.

The rest of this paper is organized as follows. Section 2 presents a brief survey of related work, followed by the details of the proposed model in Section 3. Experiment results and analysis are presented in Section 4. Finally, we conclude the paper in Section 5.

2 RELATED WORK

Our proposed method specifically tackles the conversion rate prediction problem by employing the multi-task learning framework over the entire space. Therefore, we briefly review the most related

work from the following two aspects: 1) conversion rate prediction and 2) multi-task learning.

Conversion Rate Prediction: Conversion rate prediction is a key component of many online applications, such as search engines [2, 33], recommender systems [10, 23] and online advertising [8, 12]. However, there are few literatures directly proposed for CVR tasks [16, 28, 30], regardless of recent prosperous development of CTR methods [3, 29, 34, 35]. Indeed, CVR modeling is very challenging since conversions are extremely rare events that only a very small portion of impression items are eventually being clicked and bought. Recently, the deep neural network has achieved significant progress in many areas including recommender systems due to its remarkable ability in feature representation and end-to-end modeling [4, 9, 13, 14, 17, 24]. In this paper, we also adopt a deep neural network to model the conversion rate prediction task. In contrast to the above methods, we derive a new user sequential behavior graph "impression→click→D(O)Action→purchase" based on a novel idea of post-click behavior decomposition. According to the conditional probability rule defined on the graph, our network structure is specifically devised to predict several decomposed sub-targets in parallel and compose them sequentially to formulate the final CVR.

Multi-Task Learning: Due to the temporal multi-stage nature of users' purchasing behavior, *e.g.*, impression, click, and purchase, prior work attempts to formulate the conversion rate prediction task by a multi-task learning framework. For example, Hadash *et al*. propose a multi-task learning-based recommender system by modeling the ranking and rating prediction tasks simultaneously [11]. Ma *et al*. propose a multi-task learning approach named multi-gate mixture-of-experts to explicitly learn the task relationship from data [18]. Gao *et al*. propose a neural multi-task recommendation model to learn the cascading relationship among different types of behaviors [6]. In contrast, we model the CTR and CVR tasks simultaneously by associating with users' sequential behavior graph, where the task relationship is explicitly defined by the conditional probability (See Section 3). Ni *et al*. propose to learn universal user representations across multiple tasks for more effective personalization [21]. We also explore such an idea by sharing embedded features across different tasks. Recently, Ma *et al*. propose an entire space multi-task model (ESMM) for CVR prediction [19]. It adds the CTR task and CTCVR task as an auxiliary to the main CVR task. Our method is partially inspired by ESMM but has the following significant difference: we propose a novel idea of post-click behavior decomposition to reformulate a novel user sequential behavior graph "impression→click→D(O)Action→purchase". Defining model on this graph enables to formulate the final CVR as well as some auxiliary tasks together. It can leverage all the impression samples over the entire space and the abundant supervisory signals from users' post-click behaviors, which are highly relevant to the purchase behaviors, consequently addressing the SSB and DS issue simultaneously.

3 PROPOSED METHOD

3.1 Motivation

In practice, from an item being displayed to it being purchased successfully, we identify that there may exist multiple kinds of sequential actions a user could choose to do. For example, after clicking one interested item, a user may directly purchase it without any hesitation, or add it to the shopping cart and then make the purchase eventually. These behavior paths are shown in Figure 3(a). We can simplify and group these paths according to several predefined specific purchase-related post-click actions, *i.e.*, adding to Shopping Cart (SCart) and adding to Wish list (Wish), as shown in Figure 3(b). Based on our data analysis of online real-world logs, we found that only 1% of clicked behaviors are converted to purchase eventually, indicating rare purchase training samples. However, the data volume of several post-click actions like SCart and Wish are much larger than purchase. For example, 10% will be added to the shopping cart given clicked behaviors. Besides, these post-click actions are highly relevant to the final purchase action, *e.g.*, 12% (or 31%) will be bought eventually after they have been added to the shopping cart (or wish list). How can we leverage the larger volume of post-click behaviors to benefit CVR prediction in some manner, regarding their high relevance to purchase?

Intuitively, one solution is to model these purchase-related post-click actions along with purchase into a multi-task prediction framework. The key is how to formulate them properly since they have explicit sequential correlations, *e.g.*, the purchase action probably conditioned on the SCart or Wish action. To this end, we define a single node named Deterministic Action (DAction) to merge these predefined specific purchase-related post-click actions, such as SCart and Wish, as shown in Figure 3(c). DAction has two properties: 1) it is highly relevant to the purchase action and 2) it has abundant deterministic supervisory signals from users' feedback, *e.g.*, 1 for taking some specific actions (*i.e.*, adding to shopping cart or wish list after clicking) and 0 for none. We also add a node named Other Action(OAction) between click and purchase to deal with other post-click behaviors except DAction. In this way, the conventional behavior path "impression→click→purchase" becomes to a novel elaborated user sequential behavior graph "impression→click→D(O)Action→purchase", as shown in Figure 3(c). Defining model on this graph enables to leverage all the impression samples over the entire space and extra abundant supervisory signals from D(O)Action, which will circumvent the SSB and DS issues efficiently. We call this novel idea as post-click behavior decomposition.

3.2 Conditional probability decomposition

In this section, we present the conditional probability decomposition of CVR as well as related auxiliary tasks according to the digraph defined in Figure 3(c). First, the probability of post view click-through rate of an item x_i, denoted as p_i^{ctr}, is defined as the conditional probability of being clicked given that it has been viewed, which depicts the path "impression→click" in the digraph. Mathematically, it can be written as:

$$p_i^{ctr} = p\left(c_i = 1 \,|v_i = 1\right) \triangleq y_{1i}, \tag{1}$$

where $c_i \in C$ denotes whether the i^{th} item x_i is being clicked, $c_i \in \{0, 1\}$, C is the label spaces of all the items being clicked or not, $i \in [1, N]$ and N is the number of items. Similarly, $v_i \in V$ denotes whether the i^{th} item x_i is being viewed (*i.e.*, impression), $v_i \in \{0, 1\}$, V is the label spaces of all the items being viewed or not. y_{1i} is a surrogate symbol for simplicity.

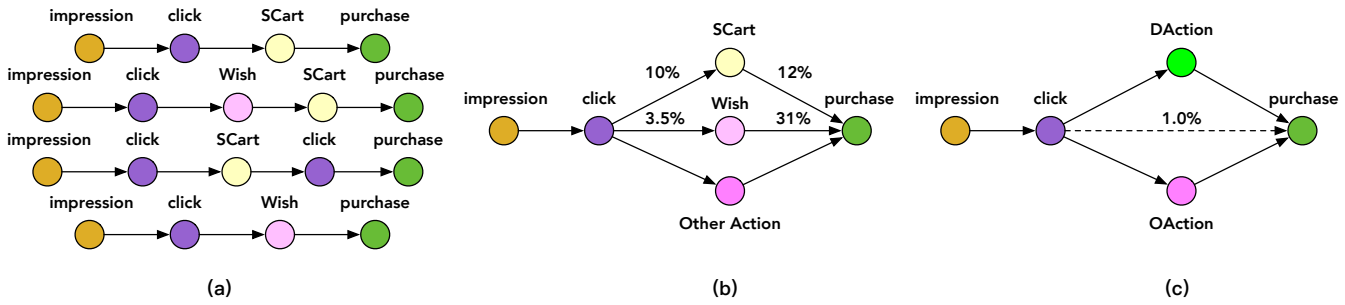

Figure 3: Illustration of the proposed user sequential behavior graph based on post-click behavior decomposition. (a) The multiple paths from impression to purchase after distinguishing post-click behaviors, such as "impression→click→SCart→purchase". (b) A digraph is used to describe the simplified purchasing process, where the numbers above edges represent the sparsity of different paths. (c) Several specific purchase-related post-click actions are merged into a single node, *i.e.*, DAction, which also inherits their supervisory signals. OAction represents other cases except DAction.

Then, the probability of click-through DAction conversion rate of an item x_i, denoted as p_i^{ctavr}, is defined as the conditional probability of being taken DAction given that it has been viewed, which depicts the path "impression→click→DAction" in the digraph. Mathematically, it can be written as:

$$\begin{aligned} p_i^{ctavr} &= p\left(a_i = 1 \,|v_i = 1\right) \\ &= \sum_{c_i \in \{0,1\}} p\left(a_i = 1 \,|v_i = 1\,, c_i\right) p\left(c_i \,|v_i = 1\right) \\ &= p\left(a_i = 1 \,|v_i = 1\,, c_i = 0\right) p\left(c_i = 0 \,|v_i = 1\right) \\ &\quad + p\left(a_i = 1 \,|v_i = 1\,, c_i = 1\right) p\left(c_i = 1 \,|v_i = 1\right) \\ &= y_{2i} y_{1i} \end{aligned}, \tag{2}$$

where $a_i \in A$ denotes whether the i^{th} item x_i is being taken some specific actions defined in Section 3.1, $a_i \in \{0, 1\}$, A is the label spaces of all the items being taken some specific actions or not. $y_{2i} = p\left(a_i = 1 \,|v_i = 1\,, c_i = 1\right)$, depicting the path "click→DAction", is a surrogate symbol for simplicity as y_{1i}. It is trivial that $y_{2i} = p\left(a_i = 1 \,|c_i = 1\right)$ since all the samples are impression samples (*i.e.*, $v_i = 1$). It is noteworthy that Eq. (2) holds due to the fact that no action occurs without being clicked, *i.e.*, $p\left(a_i = 1 \,|v_i = 1\,, c_i = 0\right) = 0$.

Next, the probability of conversion rate of an item x_i, denoted as p_i^{cvr}, is defined as the conditional probability of being bought given that it has been clicked, which depicts the paths "click→D(O)Action→purchase" in the digraph. Mathematically, it can be written as:

$$\begin{aligned} p_i^{cvr} &= p\left(b_i = 1 \,|c_i = 1\right) \\ &= \sum_{a_i \in \{0,1\}} p\left(b_i = 1 \,|c_i = 1\,, a_i\right) p\left(a_i \,|c_i = 1\right) \\ &= p\left(b_i = 1 \,|c_i = 1\,, a_i = 0\right) p\left(a_i = 0 \,|c_i = 1\right) \\ &\quad + p\left(b_i = 1 \,|c_i = 1\,, a_i = 1\right) p\left(a_i = 1 \,|c_i = 1\right) \\ &\triangleq y_{4i}\left(1 - y_{2i}\right) + y_{2i} y_{3i} \end{aligned}, \tag{3}$$

where $b_i \in B$ denotes whether the i^{th} item x_i is being bought, $b_i \in \{0, 1\}$, B is the label spaces of all the items being bought or not. $y_{3i} = p\left(b_i = 1 \,|c_i = 1\,, a_i = 1\right)$, $y_{4i} = p\left(b_i = 1 \,|c_i = 1\,, a_i = 0\right)$ are some surrogate symbols for simplicity as y_{1i}. y_{3i} or y_{4i} depicts the path "DAction→purchase" or "OAction→purchase" in the digraph, respectively.

The probability of click-through conversion rate of an item x_i, denoted as p_i^{ctcvr}, is defined as the conditional probability of being bought given that it has been viewed, which depicts the complete graph "impression→click→D(O)Action→purchase" in the digraph. Mathematically, it can be written as:

$$\begin{aligned} p_i^{ctcvr} &= p\left(b_i = 1 \,|v_i = 1\right) \\ &= \sum_{c_i} p\left(b_i = 1 \,|v_i = 1\,, c_i\right) p\left(c_i \,|v_i = 1\right) \\ &= \sum_{c_i} \sum_{a_i} p\left(b_i = 1 \,|v_i\,, c_i, a_i\right) p\left(a_i \,|v_i\,, c_i\right) p\left(c_i \,|v_i\right) \cdot \\ &= y_{4i}\left(1 - y_{2i}\right) y_{1i} + y_{3i} y_{2i} y_{1i} \\ &= y_{1i}\left(y_{4i}\left(1 - y_{2i}\right) + y_{3i} y_{2i}\right) \end{aligned} \tag{4}$$

Here, we use v_i to replace $v_i = 1$ in the third equality for simplicity without causing any ambiguity. It is noteworthy that the fourth equality holds due to the fact that no items will be bought without being clicked, *i.e.*, $p\left(b_i = 1 \,|v_i = 1\,, c_i = 0, a_i\right)$ equals to zero, $\forall a_i \in \{0, 1\}$. Indeed, Eq. (4) can be derived by decomposing the graph "impression→click→D(O)Action→purchase" into "impression→click" and "click→D(O)Action→purchase", and integrating Eq. (1) and Eq. (3) together according to the chain rule, *i.e.*, $p_i^{ctcvr} = p_i^{ctr} * p_i^{cvr}$.

3.3 Elaborated entire space supervised multi-task model

From Eq. (1)~ Eq. (4), we can see that p_i^{ctr}, p_i^{ctavr}, and p_i^{ctcvr} can be derived from four hidden probability variables y_{1i}, y_{2i}, y_{3i}, and y_{4i}, which represents the conditional probabilities over some sub-paths in the graph, *i.e.*, "impression→click", "click→DAction", "DAction→purchase" and "OAction→purchase". On the one hand, these four sub-targets are defined over the entire space and can be predicted using all the impression samples. Taking y_{2i} as an example, training y_{2i} directly with only clicked samples suffers from the SSB issue. Indeed, y_{2i} is an intermediate variable derived

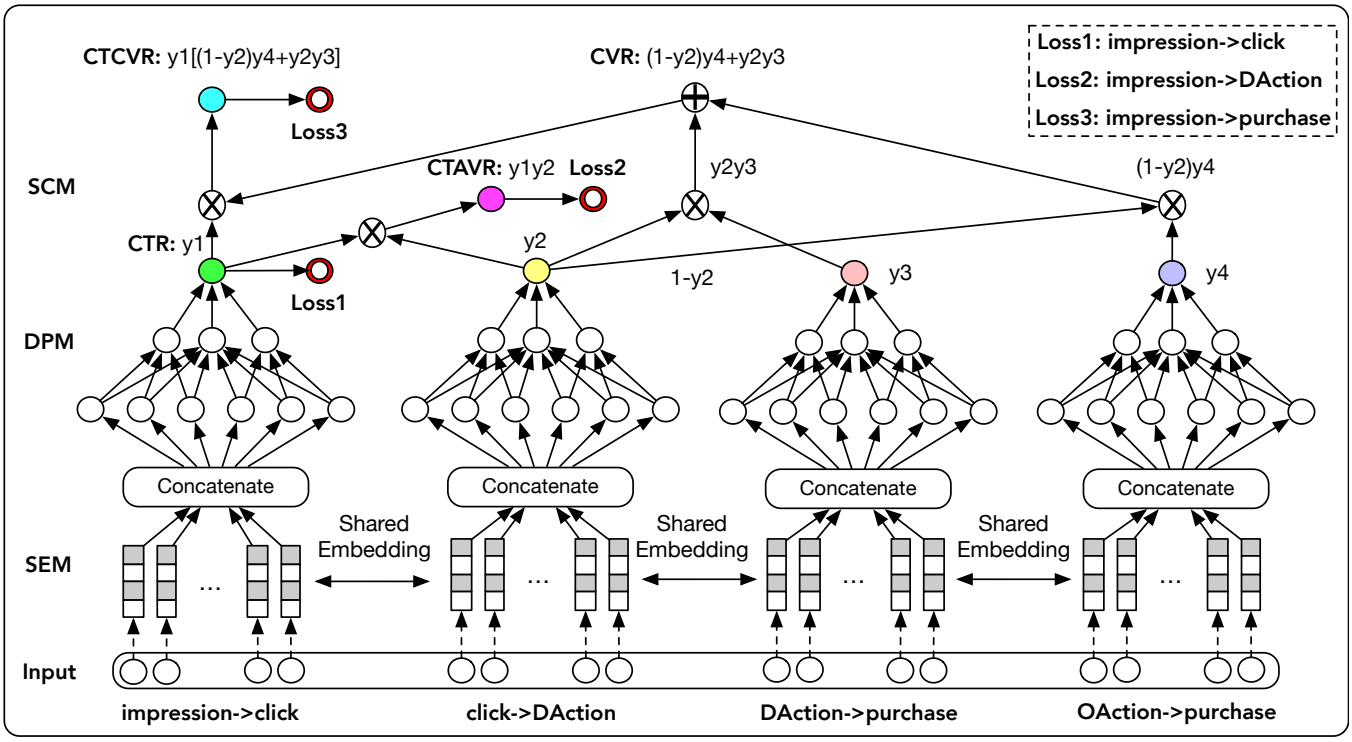

Figure 4: The diagram of ESM^2 model over the entire space, which consists of three key modules: SEM, DPM and SCM. SEM embeds sparse features into dense representation. DPM predicts the probabilities of decomposed targets. SCM integrates them together sequentially to calculate the final CVR as well as other related auxiliary tasks namely CTR, CTAVR, and CTCVR.

from p_i^{ctr} and p_i^{ctavr} according to Eq. (2). Since both p_i^{ctr} and p_i^{ctavr} are modeled over the entire space with all impression samples, the derived y_{2i} is also applicable over the entire space, therefore, no SSB in our model. On the other hand, ground truth labels of p_i^{ctr}, p_i^{ctavr}, and p_i^{ctcvr} are available given users' logs, which can be used to supervise these sub-targets. Therefore, an intuitive way is to model them simultaneously by employing a multi-task learning framework. To this end, we propose a novel deep neural recommendation model named Elaborated Entire Space Supervised Multi-task Model (ESM^2) for CVR prediction. ESM^2 gets its name since 1) p_i^{ctr}, p_i^{ctavr}, and p_i^{ctcvr} are modeled over the entire space and predicted using all the impression samples; 2) the derived p_i^{cvr} from Eq. (3) also benefits from the entire space multi-task modeling, which will be validated in the experiment part. ESM^2 consists of three key modules: 1) a shared embedding module, 2) a decomposed prediction module, and 3) a sequential composition module. We present each of them in detail as follows.

Shared Embedding Module (SEM): First, we devise a shared embedding module to embed all the sparse ID features and dense numerical features coming from user field, item field, and user-item cross field. The user features include users' ID, ages, genders and purchasing powers, *etc*. The item features include items' ID, prices, accumulated CTR and CVR from historical logs, *etc*. The user-item features include users' historical preference scores on items, *etc*. Dense numerical features are first discretized based on their boundary values and then represented as one-hot vectors. Here, we use $f_i = \{f_{ij}, \forall j \in \Lambda_f\}$ to denote the one-hot features of the i^{th} training sample, where Λ_f denotes the index set of all kinds of features. Due to the sparseness nature of one-hot encoding, we employ linear fully connected layers to embed them into dense representation, which can be formulated as:

$$g_{ij} = P_{\theta_j}^T f_{ij}, \tag{5}$$

where P_{θ_j} denotes the embedding matrix for the j^{th} kind of features, θ_j represents the network parameters.

Decomposed Prediction Module (DPM): Then, once all the feature embeddings are obtained, they are concatenated together, fed into several decomposed prediction modules, and shared by each of them. Each prediction network in DPM estimates the probability of the decomposed target on the path "impression→click", "click→DAction", "DAction→purchase", "OAction→purchase", respectively. In this paper, we employ Multi-Layer Perception (*MLP*) as the prediction network. All the non-linear activation function is *ReLU* except the output layer, where we use a *Sigmoid* function to map the output into a probability taking real value from 0 to 1. Mathematically, it can be formulated as:

$$y_{ki} = \sigma\left(\varphi_{\vartheta_k}^k(g_i)\right), \tag{6}$$

where σ denotes the *Sigmoid* function, $\varphi_{\vartheta_k}^k$ denotes the mapping function learned by the kth MLP, ϑ_k denotes its network parameters. For example, as shown in the first *MLP* in Figure 4, it output

the estimated probability y_1, which is indeed the post-view click-through rate.

Sequential Composition Module (SCM): Finally, we devise a sequential composition module to compose the above predicted probabilities sequentially according to Eq. (1) ~ Eq.(4) to calculate the conversion rate p^{cvr} and some auxiliary targets including the post-view click-through rate p^{ctr}, click-through DAction conversation rate p^{ctavr}, and click-through conversion rate p^{ctcvr}, respectively. As shown in the top part of Figure 4, SCM is a parameter-free feed forward neural network which represents the underlying conditional probabilities defined by the purchasing decision digraph in Figure 3.

Remarks: 1) All the tasks share the same embedding, making them be trained with all impression samples, *i.e.*, they are modeled over the entire space, resulting in no SSB issue during the inference phase; 2) the lightweight DPM is strictly regularized by the shared embedding module, which makes up the majority of the trainable parameters; and 3) our model suggests an efficient network design, where SEM can run in parallel, leading to low latency when deployed online.

3.4 Training objective

We use $S = \{(c_i, a_i, b_i; f_i)\}|_{i=1}^{N}$ to denote the training set, where c_i, a_i, b_i, represent the ground truth label whether the i^{th} impression sample is being clicked, taken deterministic actions, and bought. Then, we can define the joint post-view click-through probability of all training samples as follows:

$$p^{ctr} = \prod_{i \in C_+} p_i^{ctr} \prod_{j \in C_-} \left(1 - p_j^{ctr}\right), \tag{7}$$

where C_+ and C_- denote the positive and negative samples in the label space C, respectively. After taking negative logarithm on Eq.(7), we obtain the *logloss* of p^{ctr}, which is widely used in recommender systems, *i.e.*,

$$L_{ctr} = -\sum_{i \in C_+} log p_i^{ctr} - \sum_{j \in C_-} log\left(1 - p_j^{ctr}\right). \tag{8}$$

Similarly, we can obtain the loss function of p^{ctavr} and p^{ctcvr}as follows:

$$L_{ctavr} = -\sum_{i \in A_+} log p_i^{ctavr} - \sum_{j \in A_-} log\left(1 - p_j^{ctavr}\right), \tag{9}$$

and

$$L_{ctcvr} = -\sum_{i \in B_+} log p_i^{ctcvr} - \sum_{j \in B_-} log\left(1 - p_j^{ctcvr}\right). \tag{10}$$

The final training objective to be minimized is defined as:

$$L(\Theta) = w_{ctr} \times L_{ctr} + w_{ctavr} \times L_{ctavr} + w_{ctcvr} \times L_{ctcvr}, \tag{11}$$

where $\Theta = \left\{\theta_j, \forall j \in \Lambda_f\right\} \cup \{\vartheta_i, i = 1, 2, 3, 4\}$ denotes all the network parameters in ESM^2. w_{ctr}, w_{ctavr}, w_{ctcvr} are loss weights of L_{ctr}, L_{ctavr}, L_{ctcvr}, which are set to 1 in this paper, respectively.

Remarks: 1) Adding intermediate losses to supervise the decomposed sub-tasks can efficiently leverage the abundant labeled data from post-click behaviors, making the model less affected by the DS issue; and 2) all the losses are computed from the view of entire space modeling, effectively addressing the SSB issue.

4 EXPERIMENTS

To evaluate the effectiveness of the proposed ESM^2 model, we conducted extensive experiments on both the offline dataset collected from real-world e-commerce scenarios and online deployment. ESM^2 is compared with some representative state-of-the-art (SOTA) methods including GBDT [5], DNN [13], DNN using over-sampling idea [22] and ESMM [18]. First, we present the evaluation settings including the dataset preparation, evaluation metrics, a brief description of these SOTA methods, and the implementation details. Then, we present the comparison results and analysis. Ablation studies are presented next, followed by the performance analysis on different post-click behaviors.

4.1 Evaluation settings

4.1.1 Dataset preparation. We make the offline dataset by collecting the users' sequential behaviors and feedback logs [1] from our online e-commerce platform, which is one of the largest third-party retail platforms in the world. More than 300 million instances with user/item/user-item features and sequential feedback labels (*e.g.*, whether click, or DAction, or purchase) are filtered out. The statistics of this offline dataset are listed in Table 1. They are further divided into the disjoint training set, validation set, and test set.

Table 1: Statistics of the offline dataset.

Category	#User	#Item	#Impression
Number	13,383,415	10,399,095	326,325,042
Category	#Click	#Purchase	#DAction
Number	20,637,192	226,918	2,501,776

4.1.2 Evaluation metrics. To comprehensively evaluate the effectiveness of the proposed model and compare it with SOTA methods, we adopt three widely used metrics in recommendation and advertising system, *i.e.*, Area Under Curve (AUC), GAUC [35, 36] and F_1 score, where AUC reflecting the ranking ability.

$$AUC = \frac{1}{|S_+|\,|S_-|} \sum_{x^+ \in S_+} \sum_{x^- \in S_-} I\left(\phi\left(x^+\right) > \phi\left(x^-\right)\right), \tag{12}$$

where S_+ and S_- denote the set of positive/negative samples, respectively, $|S_+|$ and $|S_-|$ denote the number of samples in S_+ and S_-, $\phi(\cdot)$ is the prediction function, $I(\cdot)$ is the indicator function.

GAUC [36] is calculated as follows. First, all the test data are partitioned into different groups according to the individual user ID. Then, AUC is calculated in every single group. Finally, we average the weighted AUC. Mathematically, GAUC is defined as:

$$GAUC = \frac{\sum_u w_u \times AUC_u}{\sum_u w_u}, \tag{13}$$

where w_u denotes the weight for user u (set as 1 for our offline evaluations). AUC_u denotes the AUC for user u.

Moreover, F_1 score is defined as:

$$F_1 = \frac{2 \times P \times R}{P + R}, \tag{14}$$

[1] To the extent of our knowledge, there are no public datasets suited for this entire space modeling task, we will release our dataset for reproducibility and future research.

where P and R denote the precision and recall, *i.e.*,:

$$P = \frac{TP}{TP + FP}, \tag{15}$$

$$R = \frac{TP}{TP + FN}, \tag{16}$$

where TP, FP, and FN denote the number of true positive, false positive, and false negative predictions, respectively.

4.1.3 Brief description of comparison methods. The representative state-of-the-art methods are described as follows.

• **GBDT** [5]: The gradient boosting decision tree (GBDT) model follows the idea of gradient boosting machine (GBM), is able to produce competitive, highly robust, and interpretable procedures for regression and classification tasks [28]. In this paper, we use it as the representative of non-deep learning-based methods.

• **DNN** [13]: We also implement a deep neural network baseline model, which has the same structure and hyper-parameters with the single branch in ESM^2. Different from ESM^2, it is trained with samples on the path "click→purchase" or "impression→click" to predict conversion rate p^{cvr} or click-through rate p^{ctr}, respectively.

• **DNN-OS** [22]: Due to the data sparsity on the paths "impression→purchase" and "click→purchase", it is hard to train a deep neural network with good generalization. To address this issue, we leverage the *over-sampling* strategy to augment positive samples during training the deep model, named DNN-OS. It has the same structure and hyper-parameters with the above DNN model.

• **ESMM** [19]: For a fair comparison, we use the same backbone structure as the above deep models for ESMM. It directly models the conversion rate on the user sequential path "impression→click→purchase" without considering the purchase-related post-click behaviors.

In a nutshell, the first three methods learn to predict p^{ctr} and p^{cvr} using samples on the path "impression→click" and "click→purchase", respectively, then multiply them together to derive the click-through conversion rate p^{ctcvr}. As for ESMM and our ESM^2, they directly predict p^{ctcvr} and p^{cvr} by modeling them over the entire space.

4.1.4 Hyper-parameters settings. For the GBDT model, the number of trees, depth, minimum instance numbers for splitting a node, the sampling rate of the training set for each iteration, the sampling rate of features for each iteration, and the type of loss function, are set as 150, 8, 20, 0.6, 0.6, and *logistic loss*, respectively, which are chosen according to the AUC score on the validation set. For the deep neural network-based models, they are implemented in TensorFlow using the Adam optimizer. The learning rate is set to 0.0005 and the mini-batch size is set to 1000. Logistic loss is used as the loss function for each prediction task in all models. There are 5 layers in the *MLP*, where the dimension of each layer is set to 512, 256, 128, 32, and 2, respectively, as summarized in Table 2.

4.2 Main results

4.2.1 Comparison on the offline dataset. In this subsection, we report the AUC, GAUC, and F_1 scores of all the competitors on the offline test set. Table 3 summarizes the results of AUC and GAUC. It can be seen that the DNN method achieves gains of 0.0242, 0.0102, 0.0117 for CVR AUC, CTCVR AUC, and CTCVR GAUC over the baseline GBDT model, respectively. It demonstrates the strong representation ability of deep neural networks. Different from the vanilla DNN, DNN-OS utilizes an over-sampling strategy to address the DS issue, achieving a better performance than DNN. As for ESMM, it is modeled on the path "impression→click→purchase", which tries to address the SSB and DS issues simultaneously. Benefiting from modeling over the entire space and the abundant training samples, it outperforms DNN-OS. Nevertheless, ESMM, neglecting the impact of post-click behaviors while being further exploited by the proposed ESM^2, still struggles to address the DS issue due to rare purchase training samples. After predicting some decomposed sub-targets in parallel under a multi-task learning framework, ESM^2 composes them sequentially to formulate the final CVR. As can be seen, it obtains the best scores among all the methods. For example, the gains over ESMM are 0.0088, 0.0101, 0.0145 for CVR AUC, CTCVR AUC, and CTCVR GAUC, respectively. It is worth mentioning that a gain of 0.01 in offline AUC always means a significant increment in revenue for online RS [19, 28].

Table 2: Hyper-parameters of deep neural network-based models including DNN, DNN-OS, ESMM, and ESM^2.

Hyper-parameter	Choice
Loss function	Logistic Loss
Optimizer	Adam
Number of layers in MLP	5
Dimensions of layers in MLP	[512,256,128,32,2]
Batch size	1000
Learning rate	0.0005
Dropout ratio	0.5

Table 3: The AUC and GAUC scores of all methods.

Method	CVR AUC	CTCVR AUC	CTCVR GAUC
GBDT	0.7823	0.8059	0.7747
DNN	0.8065	0.8161	0.7864
DNN-OS	0.8124	0.8192	0.7893
ESMM	0.8398	0.8270	0.7906
ESM^2	**0.8486**	**0.8371**	**0.8051**

As for the F_1 score, we report several values by setting different thresholds for CVR and CTCVR, respectively. First, we sort all the instances in descending order according to the predicted CVR or CTCVR score. Then, due to the sparsity of CVR task (about 1% of the predicted samples are positive), we choose three thresholds namely top@0.1%, top@0.6%, and top@1% to split the predictions into positive and negative groups accordingly. Finally, we calculate the precision, recall, and F_1 scores of these predictions at these different thresholds. Results are summarized in Table 4 and Table 5. A similar trend to Table 3 can be observed. Again, the proposed method ESM^2 achieves the best performance in different settings.

4.2.2 Comparison on online deployment. It is not an easy job to deploy deep network models in our recommender system since it servers hundreds of millions of users every day, *e.g.*, more than 100 million users per second at a traffic peak. Therefore,

Table 4: The Precision, Recall and F_1 scores of all methods for CVR.

	CVR@top0.1%			CVR@top0.6%			CVR@top1%		
Method	Recall	Precision	F1-Score	Recall	Precision	F1-Score	Recall	Precision	F1-Score
GBDT	4.382%	14.348%	6.714%	16.328%	9.894%	12.322%	27.384%	7.384%	11.631%
DNN	4.938%	15.117%	7.445%	17.150%	10.495%	13.021%	28.481%	8.196%	12.729%
DNN-OS	5.383%	15.837%	8.034%	17.381%	10.839%	13.353%	29.032%	8.423%	13.058%
ESMM	5.813%	16.295%	8.570%	18.585%	11.577%	14.267%	29.789%	8.961%	13.777%
ESM^2	**6.117%**	**17.145%**	**9.017%**	**23.492%**	**10.574%**	**14.584%**	**30.032%**	**9.034%**	**13.890%**

Table 5: The Precision, Recall and F_1 scores of all methods for CTCVR.

	CTCVR@top0.1%			CTCVR@top0.6%			CTCVR@top1%		
Method	Recall	Precision	F1-Score	Recall	Precision	F1-Score	Recall	Precision	F1-Score
GBDT	2.937%	0.701%	1.132%	4.870%	0.649%	1.145%	8.894%	0.531%	1.002%
DNN	3.168%	0.851%	1.341%	5.269%	0.768%	1.340%	9.461%	0.643%	1.204%
DNN-OS	3.382%	0.871%	1.385%	5.369%	0.801%	1.395%	9.863%	0.673%	1.260%
ESMM	3.858%	0.915%	1.479%	5.504%	0.828%	1.439%	10.088%	0.691%	1.294%
ESM^2	**4.219%**	**1.001%**	**1.618%**	**5.987%**	**0.900%**	**1.566%**	**10.991%**	**0.753%**	**1.410%**

a practical model is required to make real-time CVR prediction with high throughput and low latency. For example, hundreds of recommendation items for each visitor should be predicted in less than 100 milliseconds in our system. Thanks to the parallel network structure, our model is computationally efficient and can respond to each online request within 20 milliseconds. To make the online evaluation fair, confident, and comparable, each deployed method for an A/B test has involved the same number of users, *i.e.*, millions of users. The results are listed in Figure 5, where we use the GBDT model as the baseline. As can be seen, DNN, DNN-OS, and ESMM achieve comparable performance and outperform the baseline model significantly, while ESMM performs slightly better. As for the proposed ESM^2, the significant margins between it and the above methods demonstrate its superiority. Besides, it contributes to a 3% CVR promotion compared with ESMM, indicating a significant business value for the e-commercial platform.

Remarks: 1) The deep neural network has stronger representation ability than the decision tree-based GBDT; 2) the multi-task learning framework over the entire sample space serves as an efficient tool to address the SSB and DS issues; and 3) based on the idea of post-click behaviors decomposition, ESM^2 efficiently addresses the SSB and DS issues by modeling CVR over the entire space and leveraging abundant supervisory signals from deterministic actions and achieves the best performance.

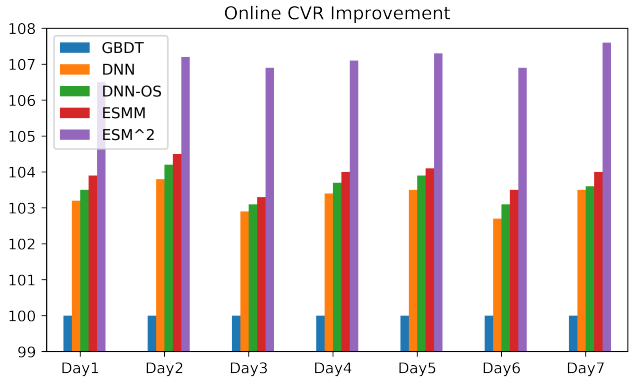

Figure 5: The results of A/B test for CVR by deploying different models in our recommender system.

4.3 Ablation studies

In this part, we present the detailed ablation studies including hyper-parameter settings of the deep neural network, effectiveness of embedding dense numerical features, and the choice of decomposing post-click behaviors, respectively.

4.3.1 Hyper-parameters of deep neural network. Here, we take three critical parameters, namely *dropout ratio*, *the number of hidden layers*, and *the dimension of item feature embeddings* as example to illustrate the process of parameter selection in our ESM^2 model.

Dropout [25] refers to the regularization technique by randomly deactivating some neural nodes during training. It can increase the generalization of the deep neural network by introducing randomness. We try different choices of the dropout ratio from 0.2 to 0.7 in our model. As shown in Figure 6(a), a dropout ratio of 0.5 leads to the best performance. Therefore, we set it to 0.5 in all the experiments if not specified.

Increasing the depth of network layers can enhance the model capacity but also potentially leads to over-fitting. Therefore, we carefully set this hyper-parameter according to the AUC scores on the validation set. As can be seen from Figure 6(b), at the beginning stage, *i.e.*, from two layers to five layers, increasing the number of hidden layers consistently improves the model's performance. However, it saturates at five layers that increasing more layers even marginally decreases the AUC scores, where the model may overfit the training set. Therefore, we use five hidden layers in all experiments if not specified.

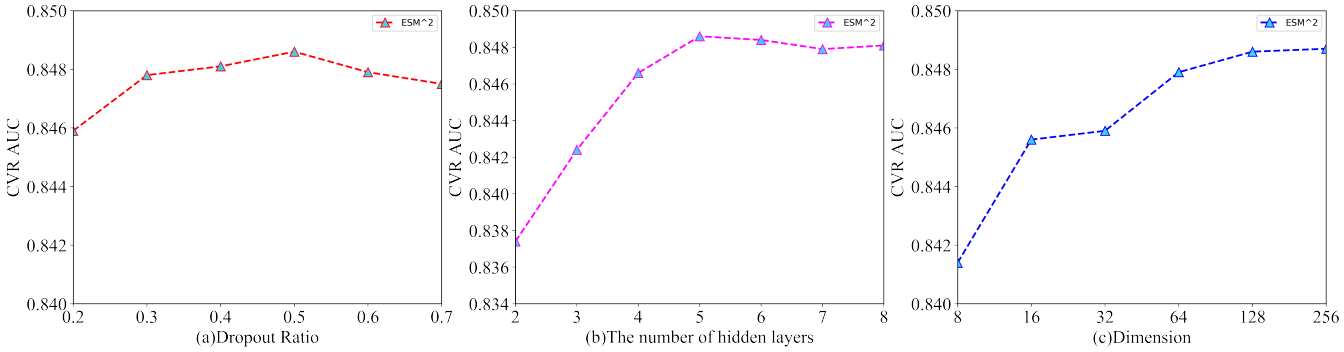

Figure 6: The results of different hyper-parameter settings in ESM^2.

The dimension of item feature embeddings is a critical parameter that high-dimension features reserve more information but also contains noise and leads to higher model complexity. We try different settings of the parameter and plot the results in Figure 6(c). As can be seen, increasing the dimension generally improves performance. It finally saturates at 128 while doubling it leads no more gains. Therefore, to make a trade-off between model capacity and complexity, we set the dimension of item feature embeddings to 128 in all the experiments if not specified.

4.3.2 Effectiveness of embedding dense numerical features. In our task, there are several numerical features. A common practice is to discretize them into one-hot vectors first and then concatenate them with ID features together, which are then embedded into dense features through a linear projection layer as described in Section 3.3. However, we hypothesize that the one-hot vector representation of numerical features may degrade the precision during discretization. In contrast, we try another solution by normalizing the numerical features and embedding them using the Tanh activation function, *i.e.*,

$$g_{ij} = \tanh\left(\frac{f_{ij} - \mu_{f_j}}{\sigma_{f_j}}\right), \tag{17}$$

where μ_{f_j} and σ_{f_j} denotes the mean and standard deviation of the j^{th} kind of features. Then, we concatenate the embedded features with the ID features together as the input of our ESM^2 model. It achieves a gain of 0.004 AUC over the discretization-based method. Therefore, we use the normalization-based embedding method for dense numerical features in all the experiments if not specified.

4.3.3 Effectiveness of decomposing post-click behaviors. When decomposing the post-click behaviors, we can integrate different behaviors into the DAction node, *e.g.*, only SCart, only Wish, or both SCart and Wish (SCart and Wish). Here, we evaluate the effectiveness of different choices. The results are summarized in Table 6. As can be seen, the combination of both SCart and Wish achieves the best AUC scores. It is reasonable since there are more purchase-related labeled data than the other two cases to address the DS issue.

Table 6: The results of choices on post-click behaviors.

	CVR AUC	CTCVR AUC	CTCVR GAUC
SCart	0.8457	0.8359	0.7996
Wish	0.8403	0.8319	0.7962
SCart and Wish	**0.8486**	**0.8371**	**0.8051**

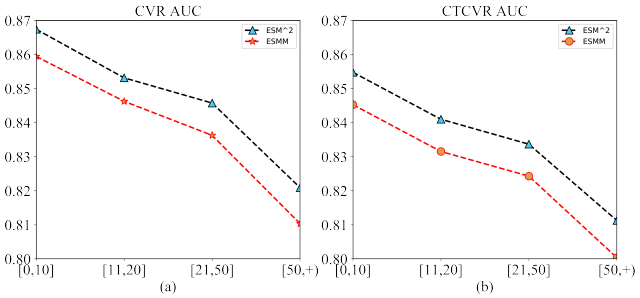

Figure 7: The AUC scores of CVR and CTCVR for ESMM and ESM^2 at different groups regarding the number of purchasing behaviors. Please refer to Section 4.4.

4.4 Performance analysis of user behaviors

To understand the performance of ESM^2 and its difference with ESMM, we further partition the test set into four groups according to the number of users' purchasing behaviors, *i.e.*, [0,10], [11,20], [21,50], [50,+). We report AUC scores of CVR and CTCVR for both methods at each group, and the results are plotted in Figure 7. As can be seen, the CVR AUC(CTCVR AUC) of both methods decreases with the increase of the number of purchasing behaviors. However, we observe that the relative gain of ESM^2 over ESMM in each group increases, *i.e.*, 0.72%, 0.81%, 1.13%, 1.30%. Generally, users having more purchasing behaviors always have more active post-click behaviors such as SCart and Wish. Our ESM^2 model deals with such post-click behaviors by adding a DAction node supervised by deterministic signals from users' feedback. Therefore, it has better representation ability on those samples than ESMM and achieves better performances on the users with high-frequency purchasing behaviors.

5 CONCLUSION

In this paper, we introduce a novel idea of post-click behavior decomposition for modeling CVR task in the context of e-commerce recommender system. A novel user sequential behavior graph "impression→click→D(O)Action→purchase" is constructed, which is used to model CVR over the entire space. Based on the conditional probability rule, we disentangle CVR and some related auxiliary tasks including the post-view click-through rate, click-through DAction conversation rate, and click-through conversion rate into four hidden probability variables, which are defined on explicit sub-paths of the graph. Consequently, we propose a novel deep neural recommendation model named ESM^2 by employing a multi-task learning framework to predict CVR as well as related auxiliary tasks simultaneously. By training with all impression samples and leveraging the abundant labels of deterministic post-click behaviors, our ESM^2 model efficiently addresses the SSB and DS issues. Extensive experiments on both offline and online environments demonstrate the superiority of ESM^2 over state-of-the-art models.

ACKNOWLEDGMENT

This work was partly supported by the National Natural Science Foundation of China (NSFC) under Grant 61806062.

REFERENCES

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-commerce Recommendation in Alibaba. *arXiv preprint arXiv:1905.06874* (2019).

[2] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 331–338.

[3] Muhammad Junaid Effendi and Syed Abbas Ali. 2017. Click through rate prediction for contextual advertisment using linear regression. *arXiv preprint arXiv:1701.08744* (2017).

[4] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. *arXiv preprint arXiv:1905.06482* (2019).

[5] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[6] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. 2019. Neural Multi-Task Recommendation from Multi-Behavior Data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1554–1557.

[7] Jennifer Golbeck, James Hendler, et al. 2006. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, Vol. 96. Citeseer, 282–286.

[8] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. Omnipress.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

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

[11] Guy Hadash, Oren Sar Shalom, and Rita Osadchy. 2018. Rank and rate: multi-task learning for recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 451–454.

[12] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 1–9.

[13] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[15] Kuang-chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. 2012. Estimating conversion rate in display advertising from past erformance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 768–776.

[16] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD'17*. 1–9.

[17] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2635–2643.

[18] 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*. ACM, 1930–1939.

[19] 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*. ACM, 1137–1140.

[20] Jeff Naruchitparames, Mehmet Hadi Güneş, and Sushil J Louis. 2011. Friend recommendations in social networks using genetic algorithms and network topology. In *2011 IEEE Congress of Evolutionary Computation (CEC)*. 2207–2214.

[21] 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*. ACM, 596–605.

[22] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 502–511.

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

[24] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM, 101–110.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[26] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *arXiv preprint arXiv:1904.06690* (2019).

[27] Gary M Weiss. 2004. Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter* 6, 1 (2004), 7–19.

[28] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2019. Multi-Level Deep Cascade Trees for Conversion Rate Prediction in Recommendation System. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[29] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[30] Hongxia Yang, Quan Lu, Angus Xianen Qiu, and Chun Han. 2016. 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*. 103–119.

[31] Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the international conference on Machine learning*.

[32] Weinan Zhang, Tianxiong Zhou, Jun Wang, and Jian Xu. 2016. Bid-aware gradient descent for unbiased learning with censored data in display advertising. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 665–674.

[33] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI Conference on Artificial Intelligence*.

[34] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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

[36] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized cost per click in taobao display advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[37] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning Tree-based Deep Model for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1079–1088.