OptDist: Learning Optimal Distribution for Customer Lifetime Value Prediction

Yunpeng Weng*
edwinweng@tencent.com
FiT, Tencent
Shenzhen, Guangdong, China

Xing Tang*
xing.tang@hotmail.com
FiT, Tencent
Shenzhen, Guangdong, China

Zhenhao Xu
zenhaoxu@tencent.com
FiT, Tencent
Shenzhen, Guangdong, China

Fuyuan Lyu†
fuyuan.lyu@mail.mcgill.ca
McGill University & MILA
Montreal, Canada

Dugang Liu‡
dugang.ldg@gmail.com
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
Shenzhen, Guangdong, China

Zexu Sun†
sunzexu21@ruc.edu.cn
Renmin University of China
Beijing, China

Xiuqiang He‡
xiuqianghe@tencent.com
FiT, Tencent
Shenzhen, Guangdong, China

Abstract

Customer Lifetime Value (CLTV) prediction is a critical task in business applications, such as customer relationship management (CRM), online marketing, etc. Accurately predicting CLTV is challenging in real-world business scenarios, as the distribution of CLTV is complex and mutable. Firstly, there is a large number of users without any consumption consisting of a long-tailed part that is too complex to fit. Secondly, the small set of high-value users spent orders of magnitude more than a typical user leading to a wide range of the CLTV distribution which is hard to capture in a single distribution. Existing approaches for CLTV estimation either assume a prior probability distribution and fit a single group of distribution-related parameters for all samples, or directly learn from the posterior distribution with manually predefined buckets in a heuristic manner. However, all these methods fail to handle complex and mutable distributions. In this paper, we propose a novel optimal distribution selection model (**OptDist**) for CLTV prediction, which utilizes an adaptive optimal sub-distribution selection mechanism to improve the accuracy of complex distribution modeling. Specifically, OptDist trains several candidate sub-distribution networks in the distribution learning module (DLM) for modeling the probability distribution of CLTV. Then, a distribution selection module (DSM) is proposed to select the sub-distribution for each sample, thus making the selection automatically and adaptively. Besides, we design an alignment mechanism that connects both modules, which effectively guides the optimization. We conduct extensive experiments on both two public and one private dataset to verify that OptDist outperforms state-of-the-art baselines. Furthermore, OptDist has been deployed on a large-scale financial platform for customer acquisition marketing campaigns and the online experiments also demonstrate the effectiveness of OptDist.

CCS Concepts

• **Information systems** → **Information systems applications**.

Keywords

Customer Lifetime Value, Probabilistic Distribution, Financial Platform

ACM Reference Format:
Yunpeng Weng, Xing Tang, Zhenhao Xu, Fuyuan Lyu, Dugang Liu, Zexu Sun, and Xiuqiang He. 2024. OptDist: Learning Optimal Distribution for Customer Lifetime Value Prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679712

1 Introduction

Predicting customer lifetime value (CLTV) is a task that refers to the estimation of potential revenue a user may bring to a platform or company [20, 37]. The accurate prediction of CLTV holds significant importance in various commercial settings, such as online advertising, marketing campaigns, and customer retention strategies [17, 35, 38]. For example, CLTV is helpful for further decision-making in customer acquisition marketing campaigns with resource constraints. Specifically, we can predict CLTV for users in a specified duration on the commercial platform and put more resources into attracting high-value customers, leading to efficient and effective utilization of budgets and higher Return on Investment (ROI).

*Contributed equally
†This work was done when working at FiT, Tencent.
‡Corresponding authors

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679712

Some conventional methods can already be directly used for CLTV modeling. One is the *statistic-based* approach [9, 11–13], which assumes a probability distribution for CLTV and obtains the parameters of the statistical distribution based on historical statistical data, such as each customer's consumption frequency and so on. With the advance of deep learning, some *value-based* approaches utilize a neural network to predict the exact value of CLTV [8, 37, 39, 42]. However, *statistic-based* approaches only rely on users' history statistics without consideration of personalization. For example, historical statistics such as frequency and recency may be the same for some users, so these methods can only roughly predict the same CLTV for them. As to *value-based* approaches, most of them adopt the Mean Squared Error (MSE) as the loss function to train a regression model, which is very sensitive to the outliers in CLTV. This leads to instability in the training and degradation of the prediction performance. Therefore, these approaches cannot predict CLTV well due to the complex and mutable distribution.

Recently, many efforts have been made to deal with the distribution of CLTV. On the whole, these methods can be divided into two categories. The first category introduces a deep probabilistic model for CLTV modeling. Zero-inflated lognormal (ZILN) [37] was commonly used to predict CLTV, which employs a deep neural network to model the zero-inflated lognormal probabilistic distribution. With inputting user and item attributes, the deep learning model can predict CLTV for a particular user behavior on a specific item. However, the real-world CLTV distribution is complex, with significant differences in the distribution of different user groups. For example, we divide users on a large-scale financial platform into four user clusters based on their attributes and further illustrate the CLTV distribution of each user cluster in Fig. 1. Based on the notable distribution difference between user groups, we can conclude that utilizing one network to learn the related parameters for all users may lead to insufficient learning. Another category is dividing the training samples into several groups according to their continuous CLTV values. Multi Distribution Multi Experts (MDME) model [20] divides data examples into multiple predefined sub-distributions based on user CLTV values, and each sub-distribution further contains numerous predefined buckets. The model aims to determine which sub-distribution the user belongs to and the optimal bucket in that sub-distribution. Nevertheless, simply dividing users into several groups requires rule-based bucketing operations and involves extremely imbalanced classification errors due to numerous zero-value values and high-value users. Moreover, even with the equal-frequency bucketing operation, the CLTV distribution within the buckets might still be uneven, leading to errors in bucket-normalized bias. Despite existing efforts, an adaptive way to deal with CLTV prediction is still required.

To tackle the above limitations, we introduce a novel framework named **Opt**imal **Dist**ribution Selection (OptDist) for CLTV prediction in this paper. Inspired by the intrinsic adaptive performance on different data samples of AutoML [26, 27, 43], OptDist adopts a distribution selection network to automatically select sub-distribution parameters for each example in a differentiable manner. Specifically, instead of using one distribution for all the data examples, we train multiple candidate sub-distribution networks (SDNs) for modeling the CLTV probabilistic distribution in the distribution learning module (DLM) following a divide-and-conquer manner. Notice that each SDN concentrates on training a possible set of probabilistic distribution parameters, thus reducing the complexity of the overall CLTV modeling. Moreover, unlike existing methods that manually partition training examples into different sub-distributions, OptDist introduces a distribution selection module (DSM) that adaptively selects one of the sub-distributions for each individual training example with Gumbel-Softmax [18] operation. Therefore, at the inference stage, we can use only the optimal sub-distribution selected by the DSM for each predicted instance without creating a gap between training and inference. However, training DSM and the DLM still poses a challenge in this framework due to the different parameter sets in these two modules. We thus propose a novel alignment mechanism to address the issue, which aligns the probability output by DSM to the distribution of loss output by DLM. The main contributions are summarized as follows:

- We propose a novel end-to-end CLTV prediction framework named OptDist. Our OptDist explores multiple candidate probabilistic distributions and selects the optimal sub-distribution for each example, which can deal with the complex and mutable distribution of customer lifetime value.
- We design two modules, DLM and DSM, respectively, to learn the sub-distribution and distribution selection. We propose an alignment mechanism connected with two modules to train the framework. With two modules and an alignment mechanism, OptDist can adaptively select the optimal sub-distribution for each data example.
- We conduct comprehensive experiments on two public datasets and a private industrial dataset to verify the superiority of our proposed OptDist model over baselines. Moreover, we have employed OptDist on a large-scale financial platform for marketing campaigns. The online A/B testing results also demonstrate the effectiveness of OptDist.

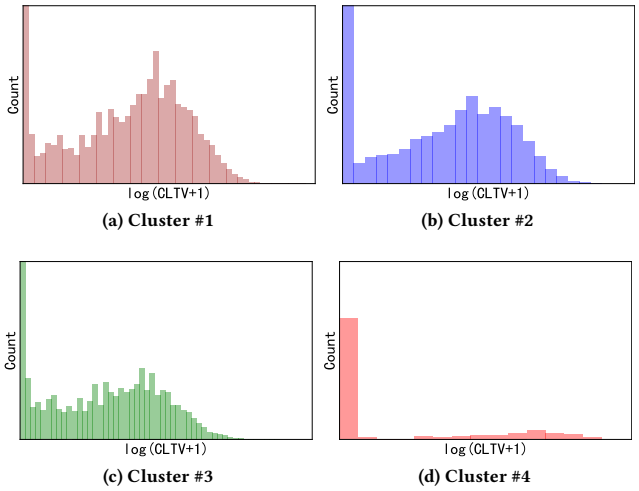


Figure 1: The logarithmic CLTV distribution of a large-scale financial platform, with four clusters representing user groups obtained by the clustering algorithm. The x-axis represents $\log(CLTV+1)$, and the y-axis shows the sample count.

2 Related work

In this section, we give a brief review of some related work. Our work is related to two topics: CLTV prediction and AutoML for the recommendation. We summarize the work in the following.

2.1 CLTV Prediction

User response modeling is essential for online marketing and recommendation systems. Traditional methods, such as click-through prediction or conversion prediction models, have shown limitations in business scenarios that aim to maximize Gross Merchandise Volume (GMV). Hence, some work [34] emphasized the necessity to estimate revenue for recommendation systems, accounting for user behavior, conversion rate, and click rate in a ranking model. To improve ROI and GMV, CLTV estimation emerged as an essential metric to evaluate commercial impact. Conventional statistic-based methods, including RFM [12] and Pareto/NBD [4, 13, 32] models, mainly focus on historical data but neglect rich user attribute information. Hence, some work incorporate the user information into the prediction model. Two-stage modeling approaches to predict both the likelihood of purchase and the value of customers are proposed in [10, 35]. Word2vec [30] is leveraged for creating user embeddings to predict CLTV [6]. Besides, some work investigated various sequential models for behaviors in CLTV [3, 39], combining RNNs with gradient boosting machines (GBMs) [3] to capture historical customer behavior, or employing wavelet transforms and attention-based GRU for analyzing user behavior sequences [39]. MarfNet [40] addresses the feature missing problem in the CLTV modeling. These works are perpendicular to our study and can potentially be combined with our method for further improvements. Introduced a prior distribution, ZILN [37] gives a multi-task solution for CLTV prediction combined classification of returning customers and prediction of returning customer spend. ExpLTV [41] further extends the ZILN to both game whale detection and CLTV prediction. Moreover, Order Dependency Monotonic Network [20] designed the order dependency monotonic network (ODMN) for modeling ordered dependencies between CLTVs to predict the value of CLTV in different periods. At each period, it uses a multi-distribution multi-expert (MDME) module that predicts the classification probabilities with pre-defined buckets and uses them to select proper experts to predict CLTV in certain ranges.

2.2 AutoML for Recommendation

In recent years, Automated Machine Learning (AutoML) techniques have gained considerable attention in the recommendation domain for their ability to automatically and efficiently find the best machine learning models and improve performance [7, 21, 46]. Most previous research focuses on parameter searching such as embedding size [28, 44, 45] and the process patterns of features including feature bucketing, interaction [15, 25]. As to embedding size search, AutoEmb [44] proposed an AutoML-driven approach that decides the optimal embedding size for user/item feature fields based on the contextual information and their popularity. AutoDim [45] is also proposed for embedding size searching, which learns multiple candidate embeddings with different embedding sizes for each feature field and uses the Gumbel-softmax trick to select the best embedding size. To achieve automated continuous feature discretization and embedding for enhancing model performance, AutoDis [15] is proposed by incorporating meta-embedding and automatic discretization modules. AutoCross [25] focused on the automatic feature interaction operation, which performs beam search in a tree-structured space and generates high-order cross features. Different from the aforementioned studies that focused on feature processing and representation learning, AutoLoss [43] focuses on the search for an appropriate loss function, which involves multiple candidate loss functions and a controller to determine their probabilities. OptDist first introduces AutoML technique to CLTV prediction, which adaptively searches the optimal distribution for data instances.

3 Method

The overall framework of OptDist is illustrated in Fig. 2. OptDist mainly consists of shared representation learning, a distribution learning module (DLM), and a distribution selection module (DSM). The shared representation learning transforms the original features into dense vectors. DLM comprises multiple sub-networks learning the parameters of one particular probabilistic distribution. The DSM contains a distribution selection network that aims to select an optimal candidate sub-distribution from DLM for each data instance. Then, we describe the alignment mechanism and how to optimize our method.

3.1 Problem Formulation

3.1.1 Customer Lifetime Value Prediction. We first give the formulation of CLTV prediction problem. Given a set of N users $\mathcal{U}$ and the total revenue a user u brings to the platform in a fixed time window d. Each sample in the training dataset $\mathcal{D} = \{(\mathbf{x}_u, y_u)|u \in \mathcal{U}, y_u \in [0, +\infty)\}$ contains the input feature $\mathbf{x}_u$ and the CLTV label $y_u \geq 0$. In general, we predict the CLTV with the model $f(\cdot)$, which can be formulated as follows,

$$\hat{y}_u = f(\mathbf{x}_u \mid \mathcal{D}, \Theta), \tag{1}$$

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{u=1}^{\mathcal{U}} \mathcal{L}_u(\hat{y}_u, y_u), \tag{2}$$

where $\hat{y}_u$ is the prediction CLTV, i.e. pLTV, Θ denotes the parameters of the model, and $\mathcal{L}$ is loss function for each users $\mathcal{L}_u$.

3.1.2 Optimal Distribution Selection. Based on Eq. 1, Θ is usually a probabilistic model that is hard to capture CLTV distribution as discussed above. We thus divided a single probabilistic model into a series of sub-distribution models $\Theta = [\theta_1, \cdots, \theta_L]$. We construct the distribution selection as learning a mask vector $\boldsymbol{\pi}_u$ for particular user and denotes $\tilde{\Theta} = \boldsymbol{\pi}_u \odot \Theta = [\pi_{u,1}\theta_1, \cdots, \pi_{u,L}\theta_L]$. With the notations defined above, our optimal distribution selection problem is formally defined as follows:

$$\theta^{\star} = \arg\min_{\boldsymbol{\pi}_u, \Theta} \mathcal{L}_u(f(\mathbf{x}_u \mid \mathcal{D}, \boldsymbol{\pi}_u \odot \Theta), y), \tag{3}$$

$$s.t. \quad \sum_{l=1}^{L} \pi_{u,l} = 1, \quad \pi_{u,l} \in \{0, 1\},$$

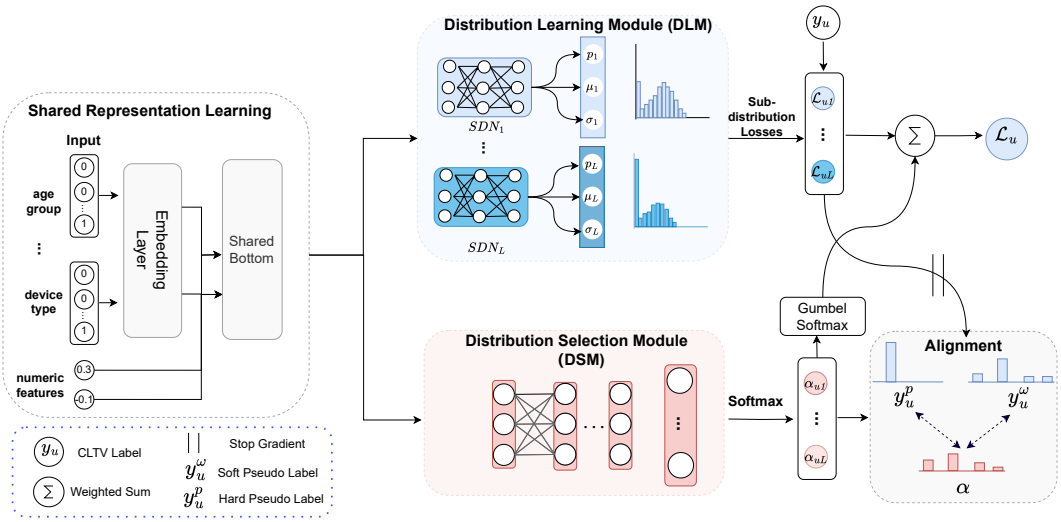


Figure 2: The overall framework of our proposed OptDist.

where L is the number of candidate sub-distributions to select and θ_ls are initialized with different seeds to increase diversity between sub-distributions.

3.2 Shared Representation Learning

The input feature $\mathbf{x}_u \in \mathcal{R}^m$ mainly consists of categorical and continuous features. Usually, each categorical feature x_i, such as the user's city or gender, will be embedded into a low dimensional vector $\mathbf{e}_i$ via the embedding table. For continuous features such as the number of visits, we normalize them using Z-score [2] and treat them as a one-dimensional vector $\mathbf{e}_j$. All feature vectors are concatenated together:

$$\mathbf{h} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m]. \tag{4}$$

Subsequently, $\mathbf{h}$ will be transformed by shared bottom layers to generate a shared sample representation and fed into the DLM and DSM, respectively. Note that $\mathbf{h}$ is input for DSM and DLM. Therefore, other representation learning modules for specific tasks can be easily plugged into our framework, such as MarfNet [40] for missing feature problems, CDAF [33] for cross-domain adaption, and so on.

3.3 Distributions Learning Module

As shown in Fig. 1, capturing the CLTV distribution within one distribution is challenging due to the large fraction of zero-consumption customers and a small number of high-value customers for most business scenarios. Therefore, we adopt the idea of *Divide-and-Conquer*, which introduces several neural networks to learn part of the distribution.

Specifically, we assume that the overall complex distribution of CLTV comprises several sub-distributions, and each user belongs to one of these sub-distributions. We use several sub-networks, denoted as sub-distribution networks (SDNs), to model each sub-distribution. Each SDN focuses on learning from a subset of users with similar distributions, thus avoiding the impact of significant distribution differences between users on the effectiveness of parameter learning. Therefore, OptDist reduces the difficulty of overall probabilistic distribution modeling. Notice that different from the method in [20], which manually divides samples into sub-distributions based on the ground truth in advance, OptDist searches the optimal sub-distribution to which each user belongs automatically.

Note that there are two critical issues in this module. First, it is crucial to determine the sub-distribution network, which indicates how to model the distribution. As is previously stated, zero-inflated lognormal distribution [37] is specially proposed for the CLTV distribution. The ZILN loss alleviates the problem of commonly used MSE loss being overly sensitive to extreme values. Therefore, we adopt the network for ZILN loss in our framework as SDNs. Second, how many neural networks will be set can decide the search space in our OptDist. In the training, each user representation $\mathbf{h}_u$ will be fed into the fixed number of candidate SDNs, obtaining a set of different ZLIN distribution parameters $\{\theta_1 = (p_1, \mu_1, \sigma_1), \theta_2 = (p_2, \mu_2, \sigma_2), ..., \theta_L = (p_L, \mu_L, \sigma_L)\}$. As a result, the search space can be N^L, which poses a significant challenge to search in a large-scale platform. Hence, there is a trade-off to determine the L. If L is too large, it will increase the burden of searching, while too small will lead to the model's inability to fit complex distributions. Thus, we set L as a hyperparameter, which can be efficiently explored in our framework. In conclusion, we calculate the negative log-likelihood loss of user u for each SDN_i:

$$\mathcal{L}_{u,i} = \begin{cases} -\log(p_{u,i}) + \log(y_u\sqrt{2\pi}\sigma_{u,i}) + \dfrac{(\log y_u - \mu_{u,i})^2}{2{\sigma_{u,i}}^2}, & C_u = 1 \\ -\log(1 - p_{u,i}), & C_u = 0 \end{cases} \tag{5}$$

where C_u denotes whether the user u is converted and y_u is the CLTV label. Then, we re-write Eq. 3 to obtain the specific loss function $\mathcal{L}_u$ by calculating the weighted sum of losses for each

SDN for each user:

$$\mathcal{L}_u = \sum_{i=1}^{L} \pi_{u,i} \cdot \mathcal{L}_{u,i}, \tag{6}$$

where $\pi_{u,i}$ is weight of user u for the i-th candidate sub-distribution. $\pi_{u,i}$ is obtained from the output of the DSM module, which is discussed in the next section.

3.4 Distribution Selection Module

Tackling the optimal distribution selection problem in Eq. 3 is challenging in our OptDist. As a potential solution, the reinforcement learning agent can only receive the reward until the optimal distribution network is selected. This prevents the direct application of reinforcement learning due to delayed rewards. Hence, to determine the weights in Eq. 6, OptDist introduces a distribution selection module following the design in [43].

Our OptDist adopts the Multi-Layer Perceptron (MLP) as the optimal distribution selection network. The output of the optimal distribution selection network is formulated as follows:

$$\boldsymbol{\alpha}_u = softmax(\text{MLP}(\mathbf{h}_u|\theta_{mlp})), \tag{7}$$

where θ_{mlp} is the parameters of MLP. It is worth noting that the selection network can be easily substituted with other more powerful models [16, 36], which is out of the scope of this paper.

However, using softmax operations might produce relatively smooth weights. This may lead to the selected SDN training being influenced by the losses of other SDNs, resulting in sub-optimal results. Gumbel-max sampling [14] is a technique that enables hard selection:

$$\begin{aligned} S_u &= \text{one\_hot}(\arg\max_i[\log \alpha_{u,i} + g_{u,i}]), \\ g_{u,i} &= -\log(-\log(U_{u,i})), \\ U_{u,i} &\sim Uniform(0,1). \end{aligned} \tag{8}$$

However, this discrete selection is non-differentiable due to the arg max operation. To tackle this, we employ the straight-through Gumbel-softmax [18]:

$$\pi_{u,i} = \frac{\exp((log(\alpha_{u,i}) + g_{u,i})/\tau)}{\sum_i \exp((log(\alpha_{u,i}) + g_{u,i})/\tau)}, \tag{9}$$

where τ is the temperature parameter, which controls the approximation degree between the Gumbel-softmax distribution and the discrete distribution. As τ approaches 0, the effect of Eq. 9 becomes closer to the arg max operation, thus getting the mask vector $\boldsymbol{\pi}_u$.

3.5 Alignment Mechanism

As in Eq. 6, we need to optimize the loss function with outputs from both modules. Specifically, the DLM module will update distribution network parameters, while DSM also updates the selection policy accordingly, making optimization difficult and sub-optimal. To further enhance the optimization, we propose an alignment mechanism inspired by meta pseudo labels [31].

In our OptDist, each SDN within the DLM module focuses on training with instances allocated to that sub-distribution by the DSM. However, two individual sets of parameters in our OptDist interfere with each other during optimization. Meanwhile, there is a lack of explicit supervised signals for the DSM, which is hard to align with the output of DLM. It is challenging to train both DSM and DLM well merely relying on the loss $\mathcal{L}_u$. By normalizing the loss values generated by different SDNs for each user, the alignment mechanism can generate pseudo labels to guide the training of the DSM, reducing the difficulty of searching for the optimal sub-distribution for DSM. As Fig. 3 illustrated, when a set of loss values on possible distribution $\mathbf{Q} = \{\mathcal{L}_{u,i}\}_{i=1}^{L}$ is given, we can obtain the hard pseudo labels $\mathbf{y}_u^p$ from these loss values:

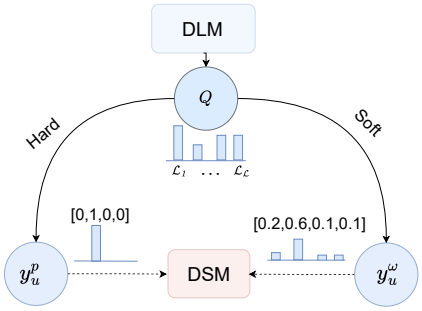


Figure 3: The alignment mechanism between DSM and DLM. Q denotes the set of SDN's losses.

$$\mathbf{y}_u^p = \text{one\_hot}(\arg\min_i(\mathcal{L}_{u,i})). \tag{10}$$

First, the hard label $\mathbf{y}^p$ can construct a cross-entropy loss. In addition, considering that in practical applications, the classification of CLTV is imbalanced, which may result in the cross-entropy of high-value users being overlooked, making it challenging for DSM to distinguish them. Therefore, to mitigate this issue, we have introduced a variant of focal weight [22] in the cross-entropy loss. The loss function can be defined as:

$$\mathcal{L}_u^{CE} = \sum_{i=1}^{L} -y_{u,i}^p (1-\alpha_{u,i})^2 \log(\alpha_{u,i}). \tag{11}$$

Then, we generate soft labels based on the losses for each sub-distribution:

$$y_u^\omega = softmax(-\mathcal{L}_u) = [\omega_{u,1}, \omega_{u,2}, ..., \omega_{u,L}] \tag{12}$$

$$\omega_{u,i} = \frac{\exp(-\mathcal{L}_{u,i})}{\sum_j \exp(-\mathcal{L}_{u,j})}. \tag{13}$$

The larger the $\omega_{u,i}$, the more suitable the i-th sub-distribution is for user u according to DLM. Then, we adopt Kullback-Leibler (KL) divergence [5] between DLM and DSM:

$$\mathcal{L}_u^{KL} = \sum_{i=1}^{L} \omega_{u,i} \log(\frac{\omega_{u,i}}{\alpha_{u,i}}). \tag{14}$$

The advantage of considering both hard and soft labels here lies in that a hard label can make DSM focus on DLM information while ignoring other label information, which is complemented by soft labels. In summary, the overall loss for OptDist is defined as:

$$\mathcal{L}^{OptDist} = \frac{1}{N} \sum_{\mathbf{u} \in \mathcal{U}} (\mathcal{L}_u + \mathcal{L}_u^{CE} + \mathcal{L}_u^{KL}). \tag{15}$$

3.6 Optimization and Inference

3.6.1 Optimization Method. In OptDist, the trainable parameters come from DLM and DSM. We denote the parameters of DLM and DSM as $\Theta_L = \{\theta_1, \cdots, \theta_L\}$ and $\Theta_S = \{\theta_{mlp}\}$, respectively. Note that $\boldsymbol{\pi}_u$ in Eq. 3 is directly generated by DSM as in Eq. 9. Here, we mainly discuss how to optimize the framework parameters. We form a bi-level optimization problem for our OptDist as follows:

$$\begin{aligned} &\min_{\Theta_S} \mathcal{L}_{val}^{OptDist}(\Theta_L^\star, \Theta_S), \\ &s.t. \quad \Theta_L^\star = \arg\min_{\Theta_L} \mathcal{L}_{train}^{OptDist}(\Theta_L, \Theta_S^\star), \end{aligned} \tag{16}$$

where DLM parameters Θ_L and DSM parameters Θ_S are considered as the upper- and lower-level variables. However, this formulation increases the complexity of model training. Therefore, We adopt an approximation scheme strategy by differentiable architecture search (DARTS) [23]. All the parameters of OptDist, denoted as $\Theta = \{\Theta_L, \Theta_S\}$, are updated as follows within each mini-batch:

$$\hat{\Theta} = \Theta - \eta \cdot \triangledown \mathcal{L}_{train}^{OptDist}, \tag{17}$$

where η is the learning rate. Note that Θ_L and Θ_S have shared and independent parameters, where the alignment mechanism alleviates the difficulty of approximating one-level optimization.

3.6.2 Inference Stage. For each instance to be predicted $\mathbf{x}$, OptDist first obtains the representation $\mathbf{h}$ through the shared embedding bottom. Then, the representation is fed into DSM, which will output the probability α that this instance belongs to each sub-distribution. Only the optimal SDN's output will be employed for the predicted CLTV calculation in this stage, and the index of that SDN could be obtained with argument max operation on α:

$$s = \arg\max([\alpha_1, \alpha_2, ..., \alpha_L]). \tag{18}$$

With the index of selected distribution, the model can fetch the optimal distribution parameters $\theta_s = (p_s, \mu_s, \sigma_s)$ from the corresponding SDN's output and combine them with the expectation formula of the log-normal distribution to obtain the estimated CLTV:

$$\hat{y} = p_s \times \exp(\mu_s + \sigma_s^2/2). \tag{19}$$

The Algorithm. 1 summarizes the optimization and inference process of OptDist. In lines 2-4, the sample representation is fed into several sub-networks to obtain the sub-distribution parameters. Since each sub-distribution only needs to handle the sub-problems of modeling the CLTV probability for a part of similar samples, the sub-network can be processed using a relatively simple MLP and can be parallelized, thus not increasing the time complexity of the model. Lines 5 and 6 present the output of DSM, while distribution evaluation is conducted from Line 7 to Line 11. Line 12 carries out the parameter update for the model. Finally, the predicted results are obtained according to Lines 13-14.

Algorithm 1: An Optimization Algorithm for OptDist

Input: Input features $\mathbf{X} = \{\mathbf{x_u} | \forall u \in \mathcal{U}\}$;
number of sub-distribution L;
temperature τ;
Initial model parameters Θ;
Learning rate η;
Output: Predicted CLTV $\hat{y}$
Trained model parameters $\hat{\Theta}$;

1 $\mathbf{H} = Emb(\mathbf{X})$;
2 **for** *i = 1 to L* **do**
3 $(p_i, \mu_i, \sigma_i) \leftarrow SDN_i(\mathbf{H})$;
4 **end**
5 α = softmax(MLP($\mathbf{H}$));
6 $\pi = gumbel\_softmax(log(\alpha), \tau)$
7 **for** *i = 1 to L* **do**
8 calculate the negative log-likelihood loss $\mathcal{L}_i$ for the corresponding sub-distribution SDN_i ;
9 **end**
10 Generate the hard label and soft label according to Eq. 10 and Eq. 13 respectively;
11 Calculate the loss $\mathcal{L}$ according to Eq. 6 - Eq. 15.
12 $\hat{\Theta} \leftarrow \Theta$ - $\eta \cdot \triangledown \mathcal{L}^{OptDist}$
13 $s = \arg\max(\alpha)$
14 $\hat{y} = p_s \times \exp(\mu_s + \sigma_s^2/2)$

4 experiment

In this section, we conduct both offline and online experiments to demonstrate the effectiveness of our proposed Optdist and answer the following research questions:

- **RQ1**: How does the offline and online performance of our proposed OptDist compare with mainstream baselines?
- **RQ2**: How do the key hyper-parameters influence the performance of OptDist?
- **RQ3**: What is the impact of the alignment mechanism in OptDist on the final result?
- **RQ4**: Can OptDist learn optimal sub-distribution?

4.1 Experimental Setup

4.1.1 Dataset. We conduct experiments on two public datasets and one private industrial dataset. In the following dataset, we randomly split them into 7:1:2 as the training, validation, and test sets, respectively.

Criteo-SSC.[1] The Criteo Sponsored Search Conversion (Criteo-SSC) Dataset is a large-scale public dataset, which contains logs obtained from Criteo Predictive Search (CPS). Each row in the dataset represents a user's click behavior on a product advertisement and contains information about the product's attributes and the user's characteristics. The label is whether the click led to a conversion and the corresponding revenue within 30 days. Note that we remove the product price from the features.

Kaggle.[2] The Kaggle's Acquire Valued Shoppers Challenge dataset, hereafter referred to as the Kaggle Dataset, contains transaction records of over 300,000 shoppers at about 3,300 companies. Similar to the experimental setting of the previous research [37], the task we consider is to predict the total value of a company's products purchased by a user in the year following their initial purchase and focus on the customers whose initial purchase occurred between

[1] https://ailab.criteo.com/criteo-sponsored-search-conversion-log-dataset/
[2] https://www.kaggle.com/c/acquire-valued-shoppers-challenge

Table 1: Dataset Statistics

Dataset	Samples	Positive Samples	Positive Ratio
Criteo-SSC	15,995,633	1,150,996	7.20%
Kaggle	805,753	726,180	90.12%
Industrial	4.535,675	287,934	6.35%

2012-03-01 and 2012-07-01. We retain the three companies with the most transactions.

Industrial. The private dataset is collected from a large-scale financial platform that offers mutual funds from various fund companies. Since customers can freely determine their investment amounts, the distribution of their customer lifetime value (CLTV) is very complex. The dataset consists of over 4.5 million samples, each corresponding to the profiles and access behavior features of a new user who has never invested in the platform. The model aims to predict whether these users will convert within the next 30 days and estimate their corresponding CLTV.

Table. 1 summarizes the details of these three datasets.

4.1.2 Metrics. The CLTV prediction is continuous, and thus, we use **MAE** to measure the deviation between the predicted value and the actual CLTV of the user, which is widely used as a metric in regression tasks [19]. In practical business applications, marketing resources tend to be allocated toward customers with higher CLTV, and thus, the accuracy of ranking users based on the predicted CLTV is more concerned [40]. Following the previous research [37, 40], we adopt both **Spearman rank correlation (Spearman's ρ)** and the **normalized Gini coefficient (Norm-GINI)** to evaluate. Notice that the larger this value, the better the CLTV prediction is. Apart from using these two ranking metrics to evaluate the overall ranking performance of the models on all samples, we also perform evaluations on positive samples separately to compare the distinguishing ability of models for non-zero CLTV samples, which are denoted as **Norm-GINI(+)** and **Spearman's** $\rho(+)$.

4.1.3 Baselines. We compared our proposed OptDist with several state-of-the-art CLTV prediction approaches. Note that some approaches focusing on representation learning [6, 39, 40] are not included here. The baselines are summarized as follows:

- Two-stage [10]. It decomposes the CLTV prediction into two tasks: the first task is a classification task predicting whether a user will churn or not, and the second task is a regression task predicting the revenue that the user brings.
- MTL-MSE [29]. It estimates conversion rate and CLTV with MSE loss according to the multi-task learning paradigm.
- ZILN [37]. ZILN assumes that the long-tailed CLTV distribution follows a zero-inflated log-normal distribution and uses a DNN to estimate the mean μ, standard deviation σ, and conversion rate p for the samples.
- MDME [20]. This baseline divides the training samples by CLTV into multiple sub-distributions and buckets, and constructs corresponding classification problems to predict the bucket a sample belongs to. In the next stage, the bias within the bucket is estimated so that the samples obtain a fine-grained CLTV value.
- MDAN [24]. MDAN predicts predefined LTV bucket labels using a multi-classification network and leverages a multi-channel learning network to derive embeddings for each bucket. The final sample representation is obtained by fusing these embeddings with the classification network's output through a weighted sum, which is then utilized for CLTV prediction.

4.1.4 Implementation Details. In this subsection, we provide the implementation details. For a fair comparison, in all experiments of OptDist and all baselines, the learning rate was chosen from [5e-4, 1e-3, 1.5e-3, 2e-3, 2.5e-3]. For both the two public datasets, the batch size was set to 2048, and the embedding size was 5. For the industrial dataset, the batch size was set to 512 and the embedding size was 12. For ZILN, MSE, and MTL-MSE, the size of the MLP part was set to [64, 64, 64] for the Kaggle dataset, [512, 256, 64] for the Criteo-SSC dataset, and [512, 256, 128] for the industrial dataset. For OptDist's each SDN, MDME's each bucket network, and MDAN's each channel network, the corresponding size was set to [64, 32, 32], [256, 128, 64], and [256, 128, 64], respectively.

Our implementation is based on Tensorflow [1] and all experiments are conducted on a Linux server with one Nvidia-Tesla V100-PCIe-32GB GPU, 128GB main memory, and 8 Intel(R) Xeon(R) Gold 6140 CPU cores. Note that the source code of the model implementation is available[3].

4.2 Performance Comparison(RQ1)

In Table 2, we present the evaluation results of each model on testing sets of all datasets, respectively. Based on these results, we have the following insightful observations:

- MTL-MSE has better performance in the overall dataset evaluation compared to the two-stage model. In evaluating the positive sample space, MTL-MSE is not necessarily superior to the two-stage models. This is because, in the two-stage methods, the learning of CLTV is more sufficient for those users with a high predicted conversion rate, and the performance of MTL-MSE might be affected by the seesaw phenomenon.
- The overall performance of ZILN is better than that of both two-stage and MTL-MSE, indicating that modeling the probability distribution of CLTV can alleviate the problem of MSE being sensitive to extreme values.
- The performance of MDME is unstable across the datasets. For example, it has a small MAE on the Criteo-SSC dataset, but both the Norm-GINI and Spearman's rank correlation metrics are poor, indicating that its ranking ability is weak on this dataset. On the Industrial dataset, although the overall Spearman's ρ is relatively better than ZILN, the overall Norm-GINI as well as the ranking metrics in the positive sample space are significantly weaker than those of ZILN and OptDist. This is because MDME needs to predict the sub-distribution and bucket to which the samples belong, as well as the position of the samples within the bucket, which might lead to the amplification of accumulated errors. Although MDAN achieves more stable results compared to MDME by fusing the embeddings of multiple channels, it still requires predefined bucketing. Additionally, it uses a single prediction network to estimate CLTV after integrating representations from

[3] https://github.com/sysuwyp/CLTV

Table 2: The overall performance of different models on all datasets. ↑ indicates that the higher value of the metric is better, while ↓ signifies the opposite. † indicates statistically significant improvement over the best baseline(p-value < 0.05).

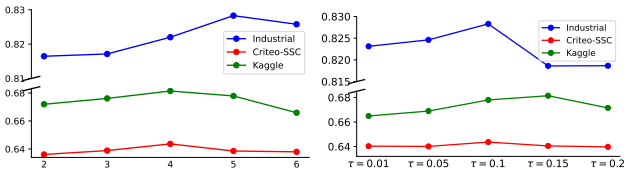
Dataset	Model	MAE ↓	Norm-GINI ↑	Spearman's ρ ↑	Norm-GINI(+) ↑	Spearman's ρ(+) ↑
Criteo-SSC	Two-stage	21.719	0.5278	0.2386	0.2204	0.2565
	MTL-MSE	21.190	0.6330	0.2478	0.4340	0.3663
	ZILN	20.880	0.6338	0.2434	0.4426	0.3874
	MDME	16.598	0.4383	0.2269	0.2297	0.2952
	MDAN	20.030	0.6209	0.2470	0.4128	0.3521
	OptDist	**15.784** †	**0.6437** †	**0.2505**	**0.4428**	**0.3903**
Kaggle	Two-stage	74.782	0.5498	0.4313	0.5505	0.4596
	MTL-MSE	74.065	0.5503	0.4329	0.5349	0.4328
	ZILN	72.528	0.6693	0.5239	0.6627	0.5303
	MDME	72.900	0.6305	0.5163	0.6213	0.5289
	MDAN	73.940	0.6648	0.4367	0.6680	0.4567
	OptDist	**70.929**†	**0.6814** †	**0.5249**	**0.6715**†	**0.5346**†
Industrial	Two-stage	0.887	0.6670	0.0781	0.5588	0.4467
	MTL-MSE	0.548	0.7194	0.1161	0.5575	0.4274
	ZILN	0.389	0.7854	0.1208	0.5899	0.5401
	MDME	0.419	0.7277	0.1229	0.5609	0.5119
	MDAN	0.437	0.7629	0.1214	0.5816	0.5383
	OptDist	**0.322**†	**0.8283** †	**0.1282**†	**0.6271**†	**0.5476**†

various channels, which may not effectively capture complex CLTV distributions, limiting the model's performance.

- Our proposed OptDist outperforms baselines across the three datasets. This indicates that by adaptively learning different sub-distribution parameters and selecting the optimal sub-distribution, it is possible to decompose the complex overall distribution into multiple relatively easy-to-learn subproblems, thereby improving the model's predictive performance. Moreover, OptDist does not require additional predefined bucketing of samples, which enables incremental training and quick deployment to other new scenarios. Due to the typically high proportion of zero-value samples in CLTV estimation problems, achieving equal frequency bucketing for MDME is difficult. Moreover, even if the positive sample bucketing is at equal frequency, the CLTV distribution within the bucket may not be uniform, leading to inaccurate bucket bias estimation.

4.3 Hyper-Parameter Sensitivity Analysis(RQ2)

In the DSM of our OptDist, Gumbel-softmax's temperature coefficient affects each SDN's weights in $\mathcal{L}_u$. Moreover, the number of sub-distributions in DLM is also a critical hyper-parameter. Therefore, this section investigates how these two hyper-parameters affect our framework. Note that we mainly focus on the Norm-GINI evaluation of the overall samples in practical business scenarios, as it indicates whether the model can help allocate marketing resources to users with the highest CLTV [40]. Therefore, concerning this metric, we mainly discuss the influence of different parameters on OptDist performance. For each dataset, we vary the number of sub-distributions in the set {2, 3, 4, 5, 6}. In general, on the one hand, a dataset with a more complex overall distribution may contain more sub-distributions, requiring more SDNs for modeling. On the other hand, an increase in the number of sub-distributions also increases the difficulty of learning with DSM. In Fig. 4 (a), we display the ranking performance of the model with different sub-distribution settings. For the Kaggle and Criteo-SSC datasets, the performance of the model is best when the number of sub-distributions is 4, while for the industrial dataset, the performance is best when the number of sub-distributions is 5. Fig. 4(b) shows the performance of the framework under different Gumbel-softmax temperature coefficients. When the temperature coefficient is too high, the sampling probability becomes smoother, and the sample cannot focus on the training of the sub-distribution it selected, thus affecting the performance.



(a) Number of Sub-distributions (b) Gumbel-softmax temperature

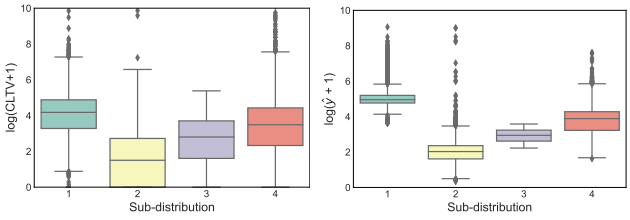
Figure 4: Norm-GINI of OptDist with different hyper-parameters on the three datasets.

4.4 Ablation Study and Case Study(RQ3&RQ4)

In this subsection, we perform the ablation experiments and case analysis to study the impact of different parts of OptDist. Firstly, we compare the OptDist with different derivations in terms of Norm-GINI of the overall samples: (1) **(w/o) Gumbel-softmax**: Remove the Gumbel-softmax operation and use the plain softmax to generate the mask vector. (2) **(w/o)** $\mathcal{L}^{KL}$: Remove the term of KL divergence loss from the alignment mechanism. (3) **(w/o)** $\mathcal{L}_u^{CE}$:

Table 3: Norm-GINI of Optdist and its derivations.

Method	Criteo-SSC	Kaggle	Industrial
OptDist	**0.6437**	**0.6814**	**0.8283**
(w/o) Gumbel-softmax	0.6389	0.6628	0.8231
(w/o) $\mathcal{L}^{KL}$	0.6382	0.6786	0.8158
(w/o) $\mathcal{L}_u^{CE}$	0.6366	0.6761	0.8107
(w/o) $\mathcal{L}^{KL}$ + $\mathcal{L}_u^{CE}$	0.6361	0.6740	0.8023



(a) Statistic of actual CLTV **(b) Statistic of Optdist prediction**

Figure 5: Difference among sub-distributions in terms of actual CLTV and predictions ($\hat{y}$) on Kaggle dataset.

Remove the term of cross-entropy from the alignment mechanism. (4) **(w/o)** $\mathcal{L}^{KL}$ + $\mathcal{L}_u^{CE}$: Omit the alignment mechanism of DSM.

We then summarize the results of ablation experiments in Table 3. Firstly, it indicates the Gumbel-softmax operation can help the OptDist improve prediction performance. Note that Gumbel-softmax is used to achieve an approximate discrete sampling and makes each SDN focus on learning from a subset of users with similar distributions. Secondly, after removing the alignment mechanism, the performance of OptDist degraded, which indicates that the alignment mechanism can effectively alleviate the training difficulty caused by the large search space. We also conduct an ablation study on both terms to investigate further the effect of KL loss and CE loss in the alignment mechanism. As it indicates, both KL divergence loss and cross-entropy loss in the alignment mechanism boost the performance, verifying our design on soft labels and hard labels. Specifically, the cross-entropy loss can make DSM training based on a guide of the best sub-distribution, and KL divergence loss ensures the DSM also takes other sub-distribution into account.

To intuitively illustrate the effectiveness of decomposing the distribution into multiple sub-distribution modeling in DLM, we conduct a case analysis on the Kaggle dataset. Fig. 5 visualizes the distributions of users in each sub-distribution in terms of actual CLTV and prediction by OptDist. The box plots indicate that OptDist can select the optimal distribution for each user and fit the sub-distribution, respectively.

4.5 Online A/B Testing(RQ1)

We have deployed the OptDist proposed in this paper on a large-scale financial platform to predict user CLTV on the platform and apply it to audience targeting in marketing campaigns.

Table 4: The relative improvement of our OptDist compared to baseline in terms of the ROI on different online campaigns.

Campaign ID	ROI-7	ROI-14	ROI-30
Campaign A	+8.96%	+17.31%	+21.90%
Campaign B	+9.04%	+9.83%	+12.51%
Campaign C	+6.47%	+8.68%	+11.45%
Campaign D	+14.42%	+16.53%	+19.06%

To ensure the fairness of the experiment, for each marketing campaign, we randomly take 50% of the traffic to the experimental group and the other 50% to the control group, ensuring that the two groups of users are homogeneous. Additionally, the marketing resources allocated to each group of traffic are equal. Following that, different models predict and rank the allocated potential users, and then select an equal number of target users for marketing campaigns.

Based on the aforementioned online A/B testing setup, we conducted online experiments on multiple marketing campaigns, focusing on users who had visited the platform in the past but had not made any purchases. The evaluation metric is the ROI, which is the ratio of the revenue contributed by users to the spend marketing budget. Table 4 presents the online experimental results on the large-scale financial technology platform. For each marketing campaign, we separately observe the relative improvement in ROI after 7 days, 14 days, and 30 days. The online experimental results demonstrated a significant improvement in OptDist across all marketing campaigns and observation time windows. These findings indicate the effectiveness of OptDist in real-world customer acquisition scenarios by accurately estimating the CLTV.

5 Conclusion

Accurately predicting CLTV is essential for increasing a company's revenue. In this paper, we propose a novel framework, OptDist, for CLTV prediction. OptDist learns multiple candidate probabilistic distributions in the DLM and adopts a network with Gumbel-softmax operation to generate exploring weights of each candidate distribution in DSM. Additionally, we propose an alignment mechanism that generates pseudo labels for DSM according to the losses of SDNs in the DLM and uses them to guide the training of DSM, thus making the optimization more effective. In this manner, OptDist decomposes the complex single distribution modeling problem into several relatively easier-to-learn sub-distribution modeling problems and selects the optimal sub-distribution for each user. We conducted comprehensive offline experiments on two public datasets and an industrial dataset, which demonstrated the superiority of OptDist. Furthermore, we have deployed our OptDist in real-world applications and conducted online experiments in multiple marketing campaigns, and the results consistently indicated the effectiveness of OptDist.

Acknowledgments

We thank the support of the National Natural Science Foundation of China (No.62302310).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.

[2] Luai Al Shalabi, Zyad Shaaban, and Basel Kasasbeh. 2006. Data mining: A preprocessing engine. *Journal of Computer Science* 2, 9 (2006), 735–739.

[3] Josef Bauer and Dietmar Jannach. 2021. Improved Customer Lifetime Value Prediction With Sequence-To-Sequence Learning and Feature-Based Models. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 5 (2021), 1–37.

[4] Albert C Bemmaor and Nicolas Glady. 2012. Modeling purchasing behavior with sudden "death": A flexible customer lifetime model. *Management Science* 58, 5 (2012), 1012–1021.

[5] Christopher M Bishop. [n. d.]. *Pattern recognition and machine learning*. Vol. 4. Springer.

[6] Benjamin Paul Chamberlain, Angelo Cardoso, CH Bryan Liu, Roberto Pagliari, and Marc Peter Deisenroth. 2017. Customer lifetime value prediction using embeddings. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1753–1762.

[7] Bo Chen, Xiangyu Zhao, Yejing Wang, Wenqi Fan, Huifeng Guo, and Ruiming Tang. 2024. A Comprehensive Survey on Automated Machine Learning for Recommendations. *ACM Trans. Recomm. Syst.* 2, 2, Article 13 (apr 2024), 38 pages.

[8] Pei Pei Chen, Anna Guitart, Ana Fernández del Río, and Africa Periánez. 2018. Customer lifetime value in video games using deep learning and parametric models. In *2018 IEEE international conference on big data (big data)*. IEEE, 2134–2140.

[9] Richard Colombo and Weina Jiang. 1999. A stochastic RFM model. *Journal of Interactive Marketing* 13, 3 (1999), 2–12.

[10] Anders Drachen, Mari Pastor, Aron Liu, Dylan Jack Fontaine, Yuan Chang, Julian Runge, Rafet Sifa, and Diego Klabjan. 2018. To be or not to be... social: Incorporating simple social features in mobile game customer lifetime value predictions. In *proceedings of the australasian computer science week multiconference*. 1–10.

[11] Peter S Fader and Bruce GS Hardie. 2009. Probability models for customer-base analysis. *Journal of interactive marketing* 23, 1 (2009), 61–69.

[12] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. 2005. RFM and CLV: Using iso-value curves for customer base analysis. *Journal of marketing research* 42, 4 (2005), 415–430.

[13] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. 2005. "Counting your customers" the easy way: An alternative to the Pareto/NBD model. *Marketing science* 24, 2 (2005), 275–284.

[14] Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*. Vol. 33. US Government Printing Office.

[15] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2910–2918.

[16] 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).

[17] Bowei He, Yunpeng Weng, Xing Tang, Ziqiang Cui, Zexu Sun, Liang Chen, Xiuqiang He, and Chen Ma. 2024. Rankability-enhanced Revenue Uplift Modeling Framework for Online Marketing. *arXiv preprint arXiv:2405.15301* (2024).

[18] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[19] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–21.

[20] Kunpeng Li, Guangcui Shao, Naijun Yang, Xiao Fang, and Yang Song. 2022. Billion-user Customer Lifetime Value Prediction: An Industrial-scale Solution from Kuaishou. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3243–3251.

[21] Yujun Li, Xing Tang, Bo Chen, Yimin Huang, Ruiming Tang, and Zhenguo Li. 2023. AutoOpt: Automatic Hyperparameter Scheduling and Optimization for Deep Click-through Rate Prediction. In *Proceedings of the 17th ACM Conference on Recommender Systems* (Singapore, Singapore) *(RecSys '23)*. Association for Computing Machinery, New York, NY, USA, 183–194.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* (2018).

[24] Wenshuang Liu, Guoqiang Xu, Bada Ye, Xinji Luo, Yancheng He, and Cunxiang Yin. 2024. MDAN: Multi-distribution Adaptive Networks for LTV Prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 409–420.

[25] Yuanfei Luo, Mengshuo Wang, Hao Zhou, Quanming Yao, Wei-Wei Tu, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. 2019. Autocross: Automatic feature crossing for tabular data in real-world applications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1936–1945.

[26] Fuyuan Lyu, Xing Tang, Huifeng Guo, Ruiming Tang, Xiuqiang He, Rui Zhang, and Xue Liu. 2022. Memorize, Factorize, or be Naive: Learning Optimal Feature Interaction Methods for CTR Prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 1450–1462. https://doi.org/10.1109/ICDE53745.2022.00113

[27] Fuyuan Lyu, Xing Tang, Dugang Liu, Liang Chen, Xiuqiang He, and Xue Liu. 2023. Optimizing Feature Set for Click-Through Rate Prediction. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 3386–3395. https://doi.org/10.1145/3543507.3583545

[28] Fuyuan Lyu, Xing Tang, Hong Zhu, Huifeng Guo, Yingxue Zhang, Ruiming Tang, and Xue Liu. 2022. OptEmbed: Learning Optimal Embedding Table for Click-through Rate Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 1399–1409.

[29] 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*. 1137–1140.

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[31] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. 2021. Meta Pseudo Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11557–11568.

[32] David C Schmittlein, Donald G Morrison, and Richard Colombo. 1987. Counting your customers: Who-are they and what will they do next? *Management science* 33, 1 (1987), 1–24.

[33] Hongzu Su, Zhekai Du, Jingjing Li, Lei Zhu, and Ke Lu. 2023. Cross-Domain Adaptative Learning for Online Advertisement Customer Lifetime Value Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (Jun. 2023), 4605–4613.

[34] Xing Tang, Yang Qiao, Fuyuan Lyu, Dugang Liu, and Xiuqiang He. 2024. Touch the Core: Exploring Task Dependence Among Hybrid Targets for Recommendation. *CoRR* abs/2403.17442 (2024). https://doi.org/10.48550/ARXIV.2403.17442 arXiv:2403.17442

[35] Ali Vanderveld, Addhyan Pandey, Angela Han, and Rajesh Parekh. 2016. An engagement-based customer lifetime value system for e-commerce. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 293–302.

[36] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*. 1785–1797.

[37] Xiaojing Wang, Tianqi Liu, and Jingang Miao. 2019. A deep probabilistic model for customer lifetime value prediction. *arXiv preprint arXiv:1912.07753* (2019).

[38] Yunpeng Weng, Xing Tang, Liang Chen, Dugang Liu, and Xiuqiang He. 2024. Expected Transaction Value Optimization for Precise Marketing in FinTech Platforms. *arXiv preprint arXiv:2401.01525* (2024).

[39] Mingzhe Xing, Shuqing Bian, Wayne Xin Zhao, Zhen Xiao, Xinji Luo, Cunxiang Yin, Jing Cai, and Yancheng He. 2021. Learning Reliable User Representations from Volatile and Sparse Data to Accurately Predict Customer Lifetime Value. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3806–3816.

[40] Xuejiao Yang, Binfeng Jia, Shuangyang Wang, and Shijie Zhang. 2023. Feature Missing-aware Routing-and-Fusion Network for Customer Lifetime Value Prediction in Advertising. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1030–1038.

[41] Shijie Zhang, Xin Yan, Xuejiao Yang, Binfeng Jia, and Shuangyang Wang. 2023. Out of the Box Thinking: Improving Customer Lifetime Value Modelling via Expert Routing and Game Whale Detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3206–3215.

[42] Shiwei Zhao, Runze Wu, Jianrong Tao, Manhu Qu, Minghao Zhao, Changjie Fan, and Hongke Zhao. 2023. perCLTV: A general system for personalized customer lifetime value prediction in online games. *ACM Transactions on Information Systems* 41, 1 (2023), 1–29.

[43] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, and Chong Wang. 2021. Autoloss: Automated loss function search in recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3959–3967.

[44] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, Chong Wang, Ming Chen, Xudong Zheng, Xiaobing Liu, and Xiwang Yang. 2021. Autoemb: Automated embedding dimensionality search in streaming recommendations. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 896–905.

[45] Xiangyu Zhao, Haochen Liu, Hui Liu, Jiliang Tang, Weiwei Guo, Jun Shi, Sida Wang, Huiji Gao, and Bo Long. 2020. Memory-efficient embedding for recommendations. *arXiv preprint arXiv:2006.14827* (2020).

[46] Ruiqi Zheng, Liang Qu, Bin Cui, Yuhui Shi, and Hongzhi Yin. 2023. AutoML for Deep Recommender Systems: A Survey. *ACM Trans. Inf. Syst.* 41, 4, Article 101 (mar 2023), 38 pages. https://doi.org/10.1145/3579355