Cross-Domain Adaptative Learning for Online Advertisement Customer Lifetime Value Prediction

Hongzu Su[1], Zhekai Du[1], Jingjing Li[1,2*], Lei Zhu[3], Ke Lu[1]

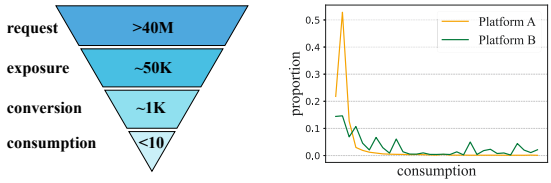
[1]University of Electronic Science and Technology of China
[2]Institute of Electronic and Information Engineering of UESTC in Guangdong
[3]Shandong Normal University
{hongzus, zhekaid}@std.uestc.edu.cn, lijin117@yeah.net, leizhu0608@gmail.com, kel@uestc.edu.cn

Abstract

Accurate estimation of customer lifetime value (LTV), which reflects the potential consumption of a user over a period of time, is crucial for the revenue management of online advertising platforms. However, predicting LTV in real-world applications is not an easy task since the user consumption data is usually insufficient within a specific domain. To tackle this problem, we propose a novel cross-domain adaptative framework (CDAF) to leverage consumption data from different domains. The proposed method is able to simultaneously mitigate the data scarce problem and the distribution gap problem caused by data from different domains. To be specific, our method firstly learns a LTV prediction model from a different but related platform with sufficient data provision. Subsequently, we exploit domain-invariant information to mitigate data scarce problem by minimizing the Wasserstein discrepancy between the encoded user representations of two domains. In addition, we design a dual-predictor schema which not only enhances domain-invariant information in the semantic space but also preserves domain-specific information for accurate target prediction. The proposed framework is evaluated on five datasets collected from real historical data on the advertising platform of Tencent Games. Experimental results verify that the proposed framework is able to significantly improve the LTV prediction performance on this platform. For instance, our method can boost DCNv2 with the improvement of 13.7% in terms of AUC on dataset G2. Code: https://github.com/TL-UESTC/CDAF.

Introduction

With the prosperity of the digital economy, a mass of entities begin to provide online services, which largely leads to the maturity of online advertisement. Among the business of online advertisement, the potential revenue from a piece of advertisement attracts the most attention of advertisers. Hence, it is crucial for the advertising platform to effectively manage customer relationship and accurately predict potential customer payment. Customer Lifetime Value (LTV) is defined as a metric that indicates how much a customer contributes to the profit during the whole lifetime relationship, which is commonly used in Marketing and Customer Relationship Management (Bauer and Jannach 2021; Xing et al. 2021). LTV prediction is therefore a promising methodology to estimate potential payment for online advertising.



(a) quantitative comparison (b) distribution comparison

Figure 1: Illustration of the data statistical results in the evaluation datasets. Figure (a) illustrates the quantitative comparison on different advertising stages and (b) illustrates the consumption distributions of two different platforms.

Currently, a variety of LTV prediction approaches have been proposed in the community. A multitude of early approaches that rely on statistics of historical customer consumption data are know as probabilistic methods since they are assumed to follow a specific probability distribution (Bauer and Jannach 2021; Borle, Singh, and Jain 2008; Khajvand et al. 2011). In real-world applications, however, the customer consumption data is volatile and complicated, which usually leads to an inferior performance of probabilistic methods. With the recent rapid development of deep learning technology, researchers propose to leverage Deep Neural Networks (DNNs) to mine the information from temporal customer data. Recent studies have verified that DNN-based methods are able to achieve more accurate LTV prediction performance than probabilistic methods (Bauer and Jannach 2021; Xing et al. 2021; Guidotti et al. 2018).

Despite the superior performance achieved in laboratory environment, previous DNN-based methods rely heavily on sufficient and reliable training samples. Unfortunately, the data of customer consumption is usually scarce and volatile in real-world online advertising scenarios. We analyze the real-world online advertising data and report the statistical results in Figure 1. As illustrated in Figure 1(a), the huge quantitative gap between advertising request and consumption indicates that collecting sufficient consumption samples

*Jingjing Li is the corresponding author. Work done when Hongzu Su was an intern at Tencent.
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

is not an easy task in reality. This problem is more severe in a newly-built advertising platform. To tackle this problem, a natural idea is to leverage consumption data from other platforms. However, there are also huge gaps between different advertising platforms. As illustrated in Figure 1(b), the discrepancy between the user consumption data distributions of two platforms indicates that utilizing data from other platforms directly may disturb the learning of the original domain, making the learned data distribution be biased to the source domain. This consequently prevents deep models from learning the inherent data distribution of the target platform like that in pure target supervised learning.

To challenge the aforementioned data scarce and distribution gap problems, we for the first time propose a cross-domain adaptative framework (CDAF) to improve the LTV prediction performance of a data-scarce platform based on user consumption data collected from another data-sufficient advertising platform. For clarity, we consider the data-scarce advertising platform as the target domain and the data-sufficient one as the source domain, casting it into a supervised domain adaptation problem (SDA). In the proposed framework, the LTV prediction model is able to simultaneously learn domain-invariant information and domain-specific information. The domain-invariant information between source and target domains refers to the consumption level and inherent characteristics of a specific user which can be leveraged to alleviate data scarce problem in the target domain. The domain-specific information refers to user consumption preference and user interaction characteristics which are able to uniquely identify a consumption distribution and be used for accurate target prediction.

Technically, we firstly train a LTV prediction model in the source domain to guide the training of the target model. We then tune this source-trained model with target domain data and simultaneously align the encoded user representations between two domains. The alignment is able to preserve domain-invariant information in a latent space by minimizing the cross-domain distribution discrepancy. To further stop the target model from over-fitting to the source domain and provide accurate target prediction, we learn domain-specific information with the specifically designed two predictors. In addition, we design a Dual Predictors Optimization (DPO) which minimizes the discrepancy of prediction distributions between two predictors to utilize both domain-invariant and domain-specific information.

The main contributions are summarized as follows: 1) We propose a novel cross-domain adaptative framework (CDAF) to mitigate data scarce and distribution gap problems in customer LTV prediction task for online advertisement. To the best of our knowledge, this is the first attempt at domain adaptive LTV prediction in the community. 2) The proposed framework is able to simultaneously learn domain-invariant and domain-specific information by minimizing Wasserstein discrepancy and optimizing the specifically designed dual predictors. 3) We conduct extensive experiments on five real-world datasets sampled from three-month historical data in the advertising platform of Tencent Games. The experimental results verify the proposed method is able to significantly improve the LTV prediction performance.

Related Work

LTV Prediction. Existing LTV prediction methods can be roughly classified into three groups: probabilistic methods, traditional machine learning methods and DNN-based methods. Probabilistic methods model the historical customer data with assumptive prior probability distributions. Among them, the most common methods are based on the negative binomial distribution. For instance, Pareto/NBD (Schmittlein, Morrison, and Colombo 1987) predicts the probability that a specific customer is active based on previous transactions. Later, BG/NBD (Fader, Hardie, and Lee 2005b,a) refines Pareto/NBD to simultaneously captures the flow of transactions and the spend of each transaction. This method has been employed to estimate the LTV of new customers on an online music site. Traditional machine learning methods commonly leverage tree-based algorithms such as Extreme Gradient Boosting (XGBoost) (Chen and Guestrin 2016). Drachen et al. (Drachen et al. 2018) leverage both Random Forest (Breiman 2001) and XGBoost algorithms to predict mobile game customer LTV with social feature involved. Besides, Vanderveld et al. (Vanderveld et al. 2016) design a model based on Random Forest with engagement features collected from a e-commerce platform to predict future value of an individual customer. Other methods such as (Chamberlain et al. 2017) also leverage Random Forest algorithm to predict LTV with tailored features or embeddings. DNN-based methods mine effective information from user features to make LTV predictions under the supervision of labeled samples. Chen et al. conducted an early work (Chen et al. 2018) which studies the application of convolutional neural networks in LTV prediction task. Some recent works such as TSUR (Xing et al. 2021) and (Bauer and Jannach 2021) propose to additionally make use of temporal information of historical customer behavior. In particular, the TSUR simultaneously learns temporal and structural user representations to accurately predict LTV.

However, all aforementioned methods require abundant labeled data for training, which can be a strong assumption in real-world scenarios (Cui et al. 2020; Lu et al. 2019). Different from them, we introduce a specifically designed cross-domain adaptative framework which leverages a data-sufficient domain to improve the LTV prediction performance of the target domain. Our feature embedding model shares a similar structure with widely used Click-Through Rate (CTR) prediction models (Wang et al. 2017; Huang, Zhang, and Zhang 2019; Wang et al. 2021) rather than the aforementioned methods because the LTV prediction model and the CTR prediction model are integrated into an advertising system to serve online advertising platforms.

Domain Adaptation generally aims to train a model in one domain (i.e, source domain) and transfer it to another domain (i.e, target domain) (Su et al. 2022). Existing domain adaptation methods can be categorized into three groups, i.e, unsupervised domain adaptation, semi-supervised / weakly-supervised domain adaptation and supervised domain adaptation, according to whether the target domain is well labeled or not. In this paper, we handle the LTV prediction task with supervised domain adaptation technology. Existing supervised domain adaptation methods are mainly focus

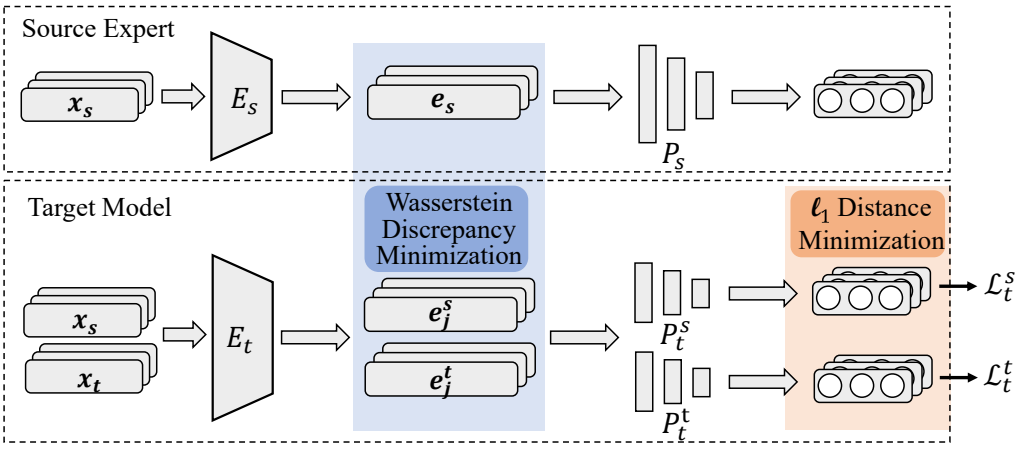


Figure 2: Illustration of our cross-domain adaptative framework. We firstly train a source expert $E_s \circ P_s$ and utilize it to initialize the target model. The target model takes user features x_s and x_t as input, and encode them into user representations e_j^s and e_j^t with the feature embedding model E_t. We explicitly align the encoded target representations with source representation e_s to learn domain-invariant information. With the dual predictors P_t^s and P_t^t, we minimize the supervised loss $\mathcal{L}_t^s$ and $\mathcal{L}_t^t$ to learn domain-specific information of source and target domain, respectively. In addition, we align the source and target predictions by minimizing ℓ_1 distance to enhance the domain-invariant information.

on minimizing the discrepancy between source and target domain data distribution with various of discrepancy metrics. Among these discrepancy metrics, the most commonly used is maximum mean discrepancy (MMD) (Gretton et al. 2012; Long et al. 2015). The MMD metric is employed to learn domain-invariant representations by comparing distributions of two domains in a latent space. Different from the MMD metric, CORAL (Sun, Feng, and Saenko 2016) introduce a novel metric which specifically aligns the second-order statistics between distributions of source and target domain. Other metrics such as Kullback–Leibler divergence are also proposed to align source and target distributions.

Proposed Method

Problem Formulation

In this paper, the platform with abundant samples is regarded as the source domain $\mathcal{S}$ and the other platform as the target domain $\mathcal{T}$. A sample of source or target domain is denoted by (x, y), where $x \in \mathcal{X}$ refers to the user feature and $y \in \mathcal{Y}$ the LTV of a consumer. We describe each domain with corresponding user feature set $\mathcal{X}$ and label set $\mathcal{Y}$, i.e., $\mathcal{S} = \{\mathcal{X}_s, \mathcal{Y}_s\}$ and $\mathcal{T} = \{\mathcal{X}_t, \mathcal{Y}_t\}$. The LTV prediction can be viewed as a regression problem which typically encodes user features x_s (resp., x_t) into corresponding user representations e_s (resp., e_t), and then predicts y_s (resp., y_t) based on the encoded e_s (resp., e_t).

Cross-Domain Adaptative Framework

As illustrated in Figure 2, the proposed method consists of two models and the training process is split into two stages: a pre-training stage and an adaptative learning stage. In the pre-training stage, a preliminary LTV prediction model (i.e., the source expert) is trained on the source domain with abundant customer consumption samples. The pre-trained source expert is able to serve as a good initialization for the target model and guides it to encode user representations according to the source knowledge. The adaptative learning stage contains two specifically designed modules: 1) **Wasserstein Discrepancy Minimization** explicitly aligns the distributions of encoded user representations between two domains by minimizing the Wasserstein distance (Lee et al. 2019; Kantorovich 2006) to learn domain-invariant information in latent space. To take advantage of both domain-invariant and domain-specific information in SDA, we design two predictors to make predictions on both source and target domains. 2) **Dual Predictors Optimization** aligns the outputs distribution between two predictors to further enhance the domain-invariant information in the output space. Finally, both of the predictors are employed to provide LTV prediction on the target domain.

Source Expert Pre-Training

In this work, we propose to utilize a source domain with abundant samples to improve the LTV prediction performance on the target domain. To this end, we firstly train an expert on the source domain. We follow the assumption that the underlying LTV data conform to lognormal distribution and optimize the LTV prediction model with a variant of ZILN loss (Wang, Liu, and Miao 2019). Formally, the source expert takes a source sample (x_s, y_s) as input and outputs prediction result (p, μ, σ). It is optimized by minimizing the following loss:

$$\begin{aligned}\mathcal{L}_s = &\mathcal{L}_{\text{CrossEntropy}}\left(\mathbb{1}_{\{y_s > N\}}; p\right) \\ &+ \mathbb{1}_{\{y_s > N\}} \mathcal{L}_{\text{Lognormal}}(y_s; \mu, \sigma),\end{aligned} \tag{1}$$

where p refers to the probability of a payment, $\mathbb{1}$ refers to the indicator function, N denotes the threshold for the indicator. The first term maximizes the probability that a cus-

tomer makes a payment. While the second term $\mathcal{L}_{\text{Lognormal}}$ maximizes the likelihood that (x_s, y_s) follows a lognormal distribution with parameters (μ, σ), which is formulated as:

$$\mathcal{L}_{\text{Lognormal}}(y_s; \mu, \sigma) = \log(y_s \sigma \sqrt{2\pi}) + \frac{(\log y_s - \mu)^2}{2\sigma^2}, \quad (2)$$

where μ and σ refer to the mean and standard deviation of lognormal distribution.

The pre-train process plays an important role in our approach: (1) It is able to serve as the feature embedding backbone to accelerate the training procedure. Similar to the commonly used backbone ResNet (He et al. 2016) and Transformer (Vaswani et al. 2017), the source expert is also trained with abundant samples and can be employed to tackle the data scarce problem. (2) The well-trained source expert is more robust than the model trained with insufficient target samples. This indicates the source expert is not prone to specific features and able to treat different user features more evenly.

Wasserstein Discrepancy Minimization

With the trained source expert, we are ready to construct the proposed adaptative framework. As illustrated in Figure 2, the target model consists of a feature embedding model E_t and two predictor models P_t^s and P_t^t, which are initialized with corresponding parameters of the source expert.

Initializing the target model with parameters of the source expert is able to alleviate the data scarce problem to some extent by source knowledge. However, the model is not able to recognize target-specific information and consequently leads to inferior prediction performance in the target domain. Fine-tuning is a straightforward methodology to solve this issue. Unfortunately, the training process in the target domain inevitably distorts the fitted feature distribution of the model which may lead to performance deterioration. To mitigate this distribution gap problem and preserve domain-invariant information, we proposed to explicitly align the encoded user representations e_s and e_t between two domains. In this work, we measure the discrepancy between e_s and e_t by Wasserstein distance which is robust and able to handle distributions that are not overlapped (Arjovsky, Chintala, and Bottou 2017). The alignment is implemented by minimizing the following Wasserstein distance:

$$\mathcal{L}_W = \inf_{\gamma \in \Pi\left(\mathbb{P}_{e_s}, \mathbb{P}_{e_t}\right)} \mathbb{E}_{(e_s, e_t) \sim \gamma} \left[\|e_s - e_t\| \right], \quad (3)$$

where $\mathbb{P}_{e_s}$ and $\mathbb{P}_{e_t}$ are marginal distributions of e_s and e_t, respectively. $\Pi\left(\mathbb{P}_{e_s}, \mathbb{P}_{e_t}\right)$ denotes all possible joint distributions of e_s and e_t. The user representations e_s and e_t are encoded as follows:

$$e_s = E_s(x_s), \; e_t = E_t(x_t), \quad (4)$$

where E_s and E_t refer to the feature embedding models of source and target domains, respectively.

As aforementioned, we design two predictors in the target model. These two predictors make predictions based on the user representations of two domains separately. Specifically, the upper one in Figure 2 is designed for source domain and the bottom one for the target domain. This structure is specifically designed to learn domain-specific information of the target domain while preserving the advantage of domain-invariant knowledge learned from the source domain. To appreciate this structure, the target model is jointly optimized by source and target data. In this case, the Eq. (3) is redefined as follows:

$$\begin{aligned} \mathcal{L}_j = & \inf_{\gamma \in \Pi\left(\mathbb{P}_{e_s}, \mathbb{P}_{e_j^s}\right)} \mathbb{E}_{(e_s, e_j^s) \sim \gamma} \left[\|e_s - e_j^s\| \right] \\ & + \inf_{\gamma \in \Pi\left(\mathbb{P}_{e_s}, \mathbb{P}_{e_j^t}\right)} \mathbb{E}_{(e_s, e_j^t) \sim \gamma} \left[\|e_s - e_j^t\| \right], \end{aligned} \quad (5)$$

where $e_j^s = E_t(x_s)$ and $e_j^t = E_t(x_t)$ denote the source representations and target representations encoded by the target feature embedding model E_t.

As a common knowledge in the community that the Wasserstein distance is difficult to be optimized (Rabin et al. 2011; Kantorovich 2006; Lee et al. 2019), we optimize Eq. (5) with the sliced Wasserstein discrepancy (SWD) (Lee et al. 2019) which measures the discrepancy on the unit sphere $\mathcal{S}^{d-1}$ in $\mathbb{R}^d$ and requires less computational cost.

Dual Predictors Optimization

As aforementioned, we design two predictors to simultaneously learn domain-specific information of target domain and preserve domain-invariant information in the source domain. The source predictor and target predictor are denoted by P_t^s and P_t^t, respectively. The prediction results are:

$$(p_s, \mu_s, \sigma_s) = P_s(e_j^s), \;\; (p_t, \mu_t, \sigma_t) = P_t(e_j^t). \quad (6)$$

We optimize the source predictor with the following loss:

$$\begin{aligned} \mathcal{L}_t^s = & \mathcal{L}_{\text{CrossEntropy}}\left(1_{\{y_s > N\}}; p_s\right) \\ & + \mathbb{1}_{\{y_s > N\}} \mathcal{L}_{\text{Lognormal}}(y_s; \mu_s, \sigma_s), \end{aligned} \quad (7)$$

where y_s refers to the LTV label corresponding to e_j^s. The target predictor is optimized by the same way:

$$\begin{aligned} \mathcal{L}_t^t = & \mathcal{L}_{\text{CrossEntropy}}\left(1_{\{y_t > N\}}; p_t\right) \\ & + \mathbb{1}_{\{y_t > N\}} \mathcal{L}_{\text{Lognormal}}(y_t; \mu_t, \sigma_t), \end{aligned} \quad (8)$$

where y_t refers to the LTV label corresponding to e_j^t.

By minimizing Eq. (7) and Eq. (8), the two predictors are able to utilize domain-specific information of source and target domain, respectively. However, different from classification problem, the output space of a regression model may change arbitrarily if data distribution varies. In this paper, we assume that the target outputs should follow a similar distribution with that of the source domain since the data-sufficient source domain is capable of describing the general LTV distribution in the real world. Technically, we propose to minimize the discrepancy between the outputs of two predictors using ℓ_1 distance (Saito et al. 2018). We therefore calculate the ℓ_1 distance as follows:

$$\mathcal{L}_d = \frac{1}{K} \sum_{i=1}^{K} |(p_s, \mu_s, \sigma_s)_i - (p_t, \mu_t, \sigma_t)_i|, \quad (9)$$

where K denotes the amount of samples, $(p_s, \mu_s, \sigma_s)_i$ and $(p_t, \mu_t, \sigma_t)_i$ refer to the prediction outputs of the i-th source and target samples, respectively. Minimizing Eq. (9) makes

the prediction of the two predictors more consistent. Thus, the two predictors are able to further preserve the shared information of two domains in the output space, i.e, the ability of leveraging domain-invariant information is enhanced.

Since the two predictors are able to simultaneously handle domain-invariant and domain-specific information, we employ both of them to provide the LTV prediction in the inference phase. The final LTV prediction result is formulated as the mean value of two predictions.

Overall Optimization Strategy

The overall optimization loss is the combination of Wasserstein discrepancy minimization, source and target supervised losses and the dual predictors optimization:

$$\mathcal{L}_a = \lambda_j \mathcal{L}_j + \lambda_s \mathcal{L}_t^s + \lambda_t \mathcal{L}_t^t + \lambda_d \mathcal{L}_d, \tag{10}$$

where λ_j, λ_s, λ_t and λ_d are hyper-parameters to balance the contribution of different components. We adopt mini-batch-based strategy to update the target model until convergence.

Experiments

In this section, we firstly detail our evaluation datasets and experimental protocols, then we conduct extensive experiments and report the experimental results to answer the following research questions: **RQ1:** How much can the cross-domain adaptative framework improve the LTV prediction performance on the target domain? **RQ2:** Does the cross-domain adaptative framework perform better than fine-tuning the pre-trained source model on the target domain? **RQ3:** How does different components of the proposed framework contribute to the LTV prediction performance? **RQ4:** How do different parameter settings affect the model performance?

Datasets Description

We evaluate our method on five real-world datasets constructed by randomly sampling from historical interaction data in two advertising platforms dedicated to games. The advertising platform with fewer samples is treated as the target domain and the other platform is treated as the source domain. We detail the statistics of evaluation datasets in Table 1. According to Table 1, the amount of samples in the source domain is 30 times larger than the target domain. The paired source and target datasets consist of samples of the same game being advertised on the source and target platforms, respectively. Both of the source and target user features consist of statistical data (such as average number of click and average consumption) and user-specific data (such as user-id, historical advertisement interaction). It is worth noting that the source user feature set is the superset of the target user feature set. We collect 400 discrete interaction data on each platform and embed them into 3200-D user features for every source and target datasets. The LTV label data is a total consumption over the last seven days provided by advertisers. The user features and LTV label are linked through anonymous advertisement identifier (such as the IDFA in Apple iOS).

Datasets	N^{train}	N^{eval}	N^{test}
G1-source	5,270,578	359,786	493,950
G1-target	68,042	4,005	4,005
G2-source	2,865,352	188,885	342,625
G2-target	44,691	3,542	3,542
G3-source	5,275,578	433,160	422,494
G3-target	140,269	11,888	11,888
G4-source	6,667,724	652,876	785,521
G4-target	2,530	5,933	5,933
G5-source	5,543,587	441,561	554,931
G5-target	183,413	12,041	12,041

Table 1: Statistics of the evaluation datasets. Notations N^{train}, N^{eval}, N^{test} refer to the amount train samples, evaluation samples and test samples, respectively.

Experimental Protocols

Evaluation Metric. In this paper, we evaluate the proposed framework with the metric of Area Under the ROC Curve (AUC) (Fawcett 2006) and normalized Gini (Wang, Liu, and Miao 2019) over the target domain. The metric of AUC is calculated by the area under a ROC curve and adopted to evaluate the accuracy of consumption probability prediction p_s and p_t. The normalized Gini is calculated by the ratio of the Gini coefficient (Gini 1997) of LTV prediction to the Gini coefficient of LTV label. This metric is used to evaluate the ability to identify high-consumption consumers from all of the consumers. High quantitative results in terms of AUC and normalized Gini indicate the outstanding predictive ability of a LTV prediction model.

Implementation Details. In our work, the LTV prediction model is composed of a feature embedding model and one or two predictor(s). Since our method is employed to improve the performance of of existing LTV prediction approaches, the feature embedding models are implemented exactly the same as reported in corresponding papers. As aforementioned, the feature embedding model in our work shares the same structure with CTR methods, and we implement them according to previous CTR prediction papers. In addition, we also implement a **Mixed** model which is a combination of GateNet and DCNv2. The predictors are implemented by three fully-connected layers with ReLU activation function. We use Adam (Kingma and Ba 2015) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize all the models. All of the feature embedding models and predictors are implemented with TensorFlow 2.4 and trained on NVIDIA Tesla V100 GPUs. All of the hyper-parameters in this work are selected with validation sets.

Quantitative Results

RQ1: Performance Comparison. To verify the proposed framework, we conduct experiments with eight feature embedding models in two experimental settings and report the experimental results in Table 2. In Table 2, experimental setting **single** refers to the model trained with only target domain data and tested on target domain. Experimental setting **CDAF** refers to the the proposed model trained with data from two domains and tested on the target domain.

Methods		G1		G2		G3		G4		G5	
		AUC	Gini	AUC	Gini	AUC	Gini	AUC	Gini	AUC	Gini
DNN (Cheng et al. 2016)	single	0.662	0.454	0.619	0.415	0.655	0.378	0.595	0.211	0.613	0.164
	CDAF	**0.675**	**0.526**	**0.662**	**0.490**	**0.730**	**0.532**	**0.766**	**0.547**	**0.718**	**0.445**
WDL (Cheng et al. 2016)	single	0.687	0.443	0.708	0.436	0.679	0.474	0.631	0.334	0.636	0.304
	CDAF	**0.720**	**0.561**	**0.714**	**0.603**	**0.722**	**0.557**	**0.788**	**0.628**	**0.713**	**0.470**
DCN (Wang et al. 2017)	single	0.671	0.403	0.694	0.579	0.560	0.107	0.628	0.341	0.612	0.162
	CDAF	**0.679**	**0.457**	**0.720**	**0.612**	**0.719**	**0.537**	**0.780**	**0.496**	**0.706**	**0.372**
DeepFM (Guo et al. 2017)	single	0.707	0.516	0.674	0.539	0.649	0.140	0.736	0.527	0.613	0.164
	CDAF	**0.726**	**0.614**	**0.713**	**0.613**	**0.759**	**0.630**	**0.790**	**0.569**	**0.718**	**0.456**
FibiNet (Huang, Zhang, and Zhang 2019)	single	0.706	0.592	0.691	0.631	0.677	0.460	0.726	0.561	0.659	0.388
	CDAF	**0.718**	**0.614**	**0.721**	**0.637**	**0.753**	**0.627**	**0.789**	**0.642**	**0.714**	**0.450**
GateNet (Huang et al. 2020)	single	0.671	0.449	0.667	0.567	0.675	0.342	0.721	0.386	0.673	0.401
	CDAF	**0.722**	**0.629**	**0.708**	**0.651**	**0.720**	**0.580**	**0.786**	**0.629**	**0.706**	**0.442**
DCNv2 (Wang et al. 2021)	single	0.663	0.337	0.633	0.428	0.611	0.261	0.726	0.435	0.653	0.283
	CDAF	**0.709**	**0.567**	**0.720**	**0.624**	**0.701**	**0.508**	**0.749**	**0.495**	**0.718**	**0.418**
Mixed Model	single	0.680	0.536	0.667	0.545	0.723	0.539	0.736	0.546	0.653	0.396
	CDAF	**0.740**	**0.625**	**0.715**	**0.640**	**0.754**	**0.639**	**0.795**	**0.689**	**0.733**	**0.461**

Table 2: Performance improvement of different models. We denote the different feature embedding models with corresponding CTR methods. The metric Gini in this table refers to the normalized Gini. The best results are marked in **Bold**.

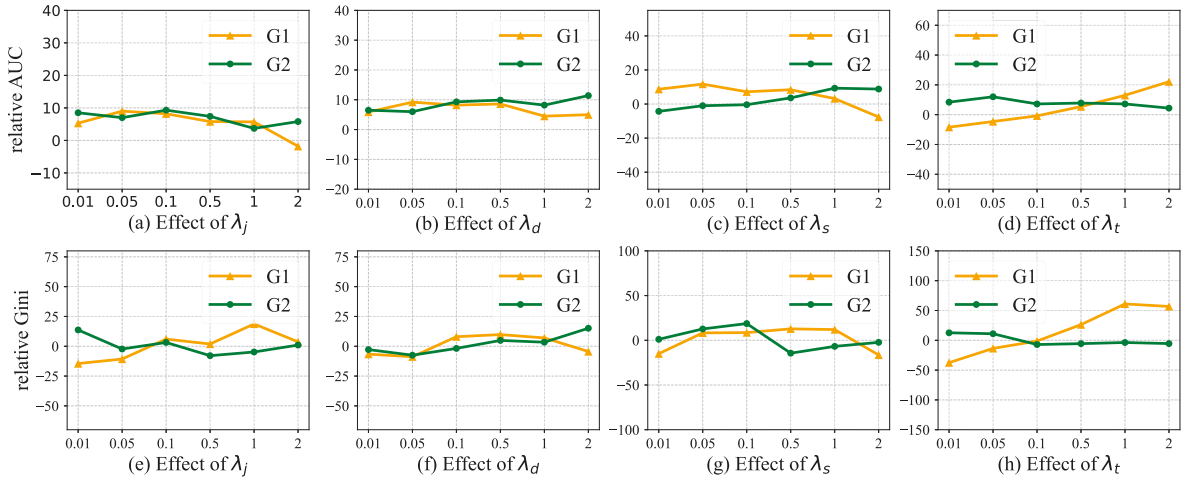


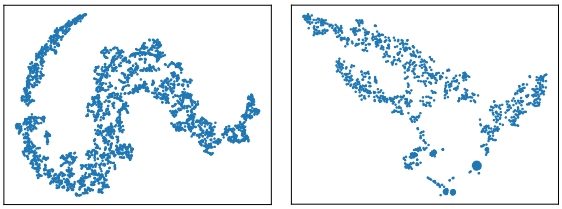
Figure 3: Parameter sensitivity analysis. We illustrate the results of Mixed model on dataset G1 and G2 as example. Figures in the first row and the second row report the results with metric $relative\ AUC$ and $relative\ Gini$, respectively.

Datasets	Fine-tune		CDAF	
	AUC	Gini	AUC (+%)	Gini (+%)
G1	0.717	0.552	**0.740** (3.2%)	**0.625** (13.2%)
G2	0.709	0.581	**0.715** (0.8%)	**0.640** (10.1%)
G3	0.696	0.572	**0.755** (8.4%)	**0.639** (11.7%)
G4	0.740	0.553	**0.795** (7.4%)	**0.690** (24.7%)
G5	0.706	0.382	**0.734** (3.9%)	**0.461** (20.6%)

Table 3: Performance comparisons between the fine-tuned model and CDAF. We report the results of Mixed model as example. The metric Gini refers to the normalized Gini.

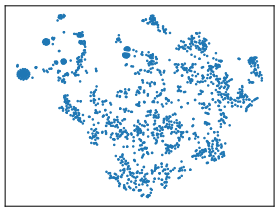
According to the results in terms of AUC, the proposed method is able to achieve the average performance improvement of 13.2%, 9.7%, 14.5%, 9.9%, 6.8%, 6.8%, 9.6% and 7.8% over five tested datasets when compared with single DNN, WDL, DCN, DeepFM, FibiNet, GateNet, DCNv2 and Mixed model, respectively. This observation indicates that the proposed cross-domain LTV prediction framework is able to mitigate the data scarcity problem in target domain and significantly improve the LTV prediction performance. We can also observe from Table 2 that our method is able to boost the prediction performance in terms of normalized Gini. For instance, the proposed method is able to surpass the single Mixed model by 16.6%, 17.4%, 18.5%, 26.1% and 16.4% on dataset G1, G2, G3, G4 and G5, respectively. This significant improvement indicates the outstanding ability to identify high-consumption users which also verify that our method is able to tackle the data scarcity problem in target domain. Another exciting observation is that the proposed framework outperforms all of the single methods listed in Table 2 which indicates our method is able to be applied to different feature embedding models to achieve performance improvement.

RQ2: Fine-tune Performance. To verify that the proposed method is able to outperforms the fine-tuned model, we con-

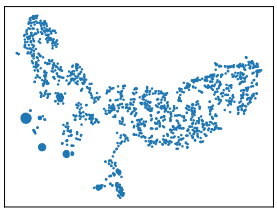


(a) Source Domain

(b) Fine-tune



(c) Target without CDAF



(d) Target with CDAF

Figure 4: Visualization results of encoded user representations for different experimental settings on G1.

Settings	G1		G2	
	AUC	Gini	AUC	Gini
CDAF w/o WD	0.726	0.610	0.714	0.622
CDAF w/o D	0.734	0.615	0.709	0.629
CDAF w/o SRC	0.731	0.590	0.704	0.580
CDAF w/o TGT	0.717	0.593	0.710	0.640
CDAF (full model)	**0.740**	**0.625**	**0.715**	**0.640**

Table 4: Results of ablation study. w/o is short for without. Notations WD, D, SRC, TGT refer to the contribution of $\mathcal{L}_j$, $\mathcal{L}_d$, $\mathcal{L}_t^s$ and $\mathcal{L}_t^t$, respectively. The metric Gini refers to the normalized Gini.

duct experiments on all of the test datasets with the implemented Mixed model and report the experimental results in Table 3. According to the results in terms of metric AUC, our method is able to surpass the fine-tuned model by 3.2%, 0.8%, 8.4%, 7.4% and 3.9% on G1, G2, G3, G4 and G5, respectively. In addition, our method achieves the improvement of 13.2%, 10.1%, 11.7%, 24.7%, 20.6% in terms of normalized Gini when compared with the fine-tuned model on G1, G2, G3, G4 and G5, respectively. This observation indicates that our method is able to handle both of the source and target data more effectively because our method is able to simultaneously learn domain-invariant information and domain-specific information.

RQ3: Ablation Study. To study the contributions of four components in our method, we conduct ablation study on G1 and G2 with the Mixed model and report the results in Table 4. According to Table 4, the contributions of four components vary between two datasets. The Wasserstein discrepancy minimization and target predictor contribute most to the improvement on G1 while the ℓ_1 distance minimization and source predictor contribute most to the improvement on G2. Another observation is that missing a component in our method does not lead to severe performance degradation. This phenomenon is caused by the specially designed framework which is able to simultaneously learn domain-invariant and domain-specific information. The Wasserstein discrepancy minimization and ℓ_1 distance minimization are able to learn domain-invariant information from two different perspectives. The source and target predictors are able to preserve domain-specific information of two domains and enhance the robustness of the full model.

RQ4: Parameter Analysis. To study the effect of the hyper-parameters corresponding to four components in our method, we conduct experiments on dataset G1 and G2 with the Mixed model and report the results in Figure 3. To better demonstrate the effect of different hyper-parameters, we report the experimental results in terms of relative AUC and relative Gini which are calculated as follows:

$$\begin{aligned} relative\ AUC &= (AUC - base_a) \times 1000, \\ relative\ Gini &= (Gini - base_g) \times 1000, \end{aligned} \tag{11}$$

where $base_a$ and $base_g$ are chosen according to the real AUC and Gini, respectively.

According to Figure 3, the effect of hyper-parameters varies between two datasets. Figure 3(a) and 3(e) indicate that the model is prone to achieve the better performance when λ_j ranges from 0.005 to 0.5. Figure 3(b) and 3(f) indicate that the model is able to perform better with greater λ_d. Hyper-parameter λ_s is supposed to be chosen on scale from 0.05 to 1.0 according to Figure 3(c) and 3(g). According to Figure 3(d) and 3(h), the Mixed model is more sensitive to hyper-parameter λ_t on dataset G1.

Qualitative Analysis

To intuitively demonstrate the ability of simultaneous learning domain-invariant and domain-specific information of our method, we collect the user representations encoded by the Mixed model on G1 and visual the encoded user representations by t-SNE (Van der Maaten and Hinton 2008). The visualization results are illustrated in Figure 4. We can observe that user representations encoded by the model without our method are chaotic which makes it difficult for the LTV predictor to learn an appropriate feature distribution. However, user representations encoded by our method present a regular distribution. In addition, we can also observe that the distribution of our method shares characteristics of both source and target distributions. This observation verifies that our method is able to simultaneously learn domain-invariant and domain-specific information.

Conclusion

In this work, we propose a cross-domain adaptative framework to tackle the data scarcity problem in the LTV prediction task. In the proposed framework, source and target user representations are specifically aligned by by minimizing the Wasserstein discrepancy to learn domain-invariant information. We design two predictors in our method to simultaneously preserve the domain-specific information and enhance the domain-invariant information. Extensive experiments over five real-world datasets verify that our method is able to significantly improve the LTV prediction performance on target domain.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62176042 and 62073059, and in part by CCF-Baidu Open Fund (NO.2021PP15002000), and in part by CCF-Tencent Open Fund (NO.RAGR20210107), and in part by Guangdong Basic and Applied Basic Research Foundation (No.2021B1515140013).

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Bauer, J.; and Jannach, D. 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): 1–37.

Borle, S.; Singh, S. S.; and Jain, D. C. 2008. Customer lifetime value measurement. *Management science*, 54(1): 100–112.

Breiman, L. 2001. Random forests. *Machine learning*, 45(1): 5–32.

Chamberlain, B. P.; Cardoso, A.; Liu, C. B.; Pagliari, R.; and Deisenroth, M. P. 2017. Customer lifetime value prediction using embeddings. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1753–1762.

Chen, P. P.; Guitart, A.; del Río, A. F.; and Periánez, A. 2018. Customer lifetime value in video games using deep learning and parametric models. In *2018 IEEE international conference on big data (big data)*, 2134–2140. IEEE.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, 7–10.

Cui, H.; Zhu, L.; Li, J.; Yang, Y.; and Nie, L. 2020. Scalable Deep Hashing for Large-Scale Social Image Retrieval. *IEEE Transactions on Image Processing*, 29: 1271–1284.

Drachen, A.; Pastor, M.; Liu, A.; Fontaine, D. J.; Chang, Y.; Runge, J.; Sifa, R.; and Klabjan, D. 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.

Fader, P. S.; Hardie, B. G.; and Lee, K. L. 2005a. RFM and CLV: Using iso-value curves for customer base analysis. *Journal of marketing research*, 42(4): 415–430.

Fader, P. S.; Hardie, B. G.; and Lee, K. L. 2005b. "Counting your customers" the easy way: An alternative to the Pareto/NBD model. *Marketing science*, 24(2): 275–284.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters*, 27(8): 861–874.

Gini, C. 1997. Concentration and dependency ratios. *Rivista di politica economica*, 87: 769–792.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773.

Guidotti, R.; Rossetti, G.; Pappalardo, L.; Giannotti, F.; and Pedreschi, D. 2018. Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Transactions on Knowledge and Data Engineering*, 31(11): 2151–2163.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1725–1731.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, T.; She, Q.; Wang, Z.; and Zhang, J. 2020. GateNet: gating-enhanced deep network for click-through rate prediction. *arXiv preprint arXiv:2007.03519*.

Huang, T.; Zhang, Z.; and Zhang, J. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 169–177.

Kantorovich, L. V. 2006. On the translocation of masses. *Journal of mathematical sciences*, 133(4): 1381–1382.

Khajvand, M.; Zolfaghar, K.; Ashoori, S.; and Alizadeh, S. 2011. Estimating customer lifetime value based on RFM analysis of customer purchase behavior: Case study. *Procedia Computer Science*, 3: 57–63.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Lee, C.-Y.; Batra, T.; Baig, M. H.; and Ulbricht, D. 2019. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10285–10295.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, 97–105. PMLR.

Lu, X.; Zhu, L.; Cheng, Z.; Nie, L.; and Zhang, H. 2019. Online Multi-modal Hashing with Dynamic Query-adaption. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 715–724.

Rabin, J.; Peyré, G.; Delon, J.; and Bernot, M. 2011. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, 435–446. Springer.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3723–3732.

Schmittlein, D. C.; Morrison, D. G.; and Colombo, R. 1987. Counting your customers: Who-are they and what will they do next? *Management science*, 33(1): 1–24.

Su, H.; Zhang, Y.; Yang, X.; Hua, H.; Wang, S.; and Li, J. 2022. Cross-domain Recommendation via Adversarial Adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1808–1817.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vanderveld, A.; Pandey, A.; Han, A.; and Parekh, R. 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.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, R.; Fu, B.; Fu, G.; and Wang, M. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, 1–7.

Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; and Chi, E. 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.

Wang, X.; Liu, T.; and Miao, J. 2019. A deep probabilistic model for customer lifetime value prediction. *arXiv preprint arXiv:1912.07753*.

Xing, M.; Bian, S.; Zhao, W. X.; Xiao, Z.; Luo, X.; Yin, C.; Cai, J.; and He, Y. 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.