aDMSCN: A Novel Perspective for User Intent Prediction in Customer Service Bots

Kuan Xu, Chilin Fu
Xiaolu Zhang, Cen Chen, Ya-Lin Zhang
{xukuan.xk,chilin.fcl,yueyin.zxl,chencen.cc,lyn.zyl}@antgroup.com
Ant Group
Hangzhou, China

Wenge Rong
w.rong@buaa.edu.cn
School of Computer Science and Engineering
Beihang University
Beijin, China

Zujie Wen, Jun Zhou*
Xiaolong Li
{zujie.wzj,jun.zhoujun,xl.li}@antgroup.com
Ant Group
Hangzhou, China

Yu Qiao
qiaoyu@sjtu.edu.cn
Department of Automation
Shanghai Jiaotong University
Shanghai, China

ABSTRACT

As one of the core components of customer service bot, User Intent Prediction (UIP) aims at predicting users' intents (usually represented as predefined user questions) before they ask, and has been widely applied in real applications. However, when developing a machine learning system for this problem, two critical issues, i.e., the problem of *feature drift* and *class imbalance*, may emerge and seriously deprave the system performance. Moreover, various scenarios may arise due to business demands, making the aforementioned problems much more severe. To address these two problems, we propose an attention-based Deep Multi-instance Sequential Cross Network (aDMSCN) to deal with the UIP task. On the one hand, the UIP task can be subtly formalized as multi-instance learning (MIL) task with an attention-based method proposed to alleviate the influences of *feature drift*. To the best of our knowledge, this is the first attempt to model the problem from a MIL perspective. On the other hand, a ratio-sensitive loss is also developed in our model, which can mitigate the negative impact of *class imbalance*. Extensive experiments on both offline real-world datasets and online A/B testing show that our proposed framework significantly outperforms other state-of-art methods for the UIP task.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

User intent prediction, Recommender system, Multiple instance learning

*Corresponding author

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

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412683

ACM Reference Format:
Kuan Xu, Chilin Fu, Xiaolu Zhang, Cen Chen, Ya-Lin Zhang, Wenge Rong, Zujie Wen, Jun Zhou, Xiaolong Li, and Yu Qiao. 2020. aDMSCN: A Novel Perspective for User Intent Prediction in Customer Service Bots. In *The 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412683

1 INTRODUCTION

As a core function of customer service bot, User Intent Prediction (UIP) [22, 25] aims to actively predict customer's intents before they ask, enable the bots to provide efficient online self-service. To achieve this purpose, UIP system usually dependents on a list of predefined questions to tag user intents [5, 6]. For example, a customer who intends to request payment related service can be potentially tagged with predefined question "why the payment failed" or "what is the payment limit". Based on that, the system predicts customers' intent through standard question recommendation, which can be handled by various Click-Through Rate (CTR) models [5, 7, 10].

Although these methods have shown promising results, there are two critical challenges, *feature drift* [2, 9] and *class imbalance* [4, 20, 27], which severely deprave model performance. The feature drift problem stems from the evolution of business scenarios. Unlike traditional recommendation tasks which focus on a single business scenario, in the online service system, the user intents may be raised from different business scenarios/domains, e.g., payment, loan, insurance, etc. In order to improve the model performance in the UIP task for all kinds of business scenarios, it is necessary to design specific features for each scenario, such as "user's latest payment channel" for the payment scenario and "whether the user has purchased insurance products" for the insurance scenario. With the evolution of business, new business scenarios may rapidly emerge along with the termination of some former businesses. As illustrated in Figure 1, the valid features for the intent prediction can change dramatically with the changing business scenarios. As a result, feature drift problem may naturally occur, where some features for a particular scenario become or cease to be relevant to the prediction task. A simple strategy sums up all available features into a aggregated representation, which pays equal attention to all features and fails to capture the critical ones.

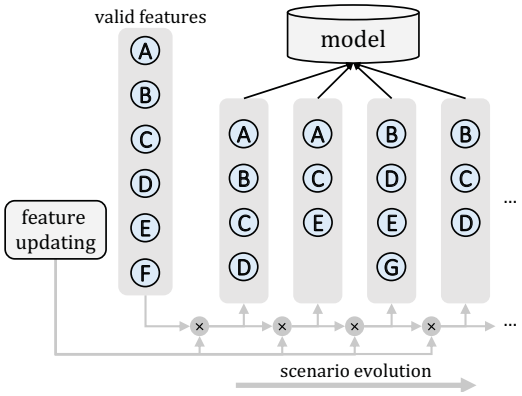

Figure 1: Illustration of feature drift in online service bots. The valid feature set changes with the evolution of business scenarios, causing inconsistency in feature quantities and categories.

Meanwhile, most methods [5] formulate UIP task as a multi-classification problem where items are considered as mutually exclusive classes. As user intents differ significantly especially for different scenarios where each intent/question is viewed as a unique class, extreme class imbalance problems may exist. In our customer service bot, there are over 10,000 unique classes under hundreds of scenarios, and 90% user intents come from 10% of these classes. Hence, the UIP model may be overwhelmed by excessive training signals from dominant classes and thus corrupt the system performance.

In this paper, we propose an attention-based Deep Multi-instance Sequential Cross Network framework (aDMSCN) to tackle the aforementioned problems in UIP tasks. Inspired by the success of MIL in web mining [35], we investigated our UIP tasks from the multi-instance perspective, where the features are treated as instances and the user depicted by these features is regarded as a bag. The predefined question interested by the user is regarded as the label of the bag. The bag representation is calculated by weighted averaging of the instances feature. A deep neural network is introduced here to determine the weights by an attention-based mechanism. The MIL setting eliminates the constraint of the fixed number of features required for most CTR prediction methods, thus alleviating the feature drift problem. Moreover, we propose a novel ratio-sensitive loss function to handle the imbalanced distribution of the classes. Unlike previous weighted loss methods that rely only on predictive or ground truth classes [20], our loss function regards the relative balance between the underlying facts and predictions, assigning sample-wise loss weights with respect to the balance ratio between the ground truth and the predictions. In addition, we record the IDs of different pages of user visits and sort them according to the access time. The resulting sequence is called the user behavior trajectory. In order to exploit the value of the user behavior trajectory, a sequential attention method is introduced by combining LSTM with an attention mechanism to better capture the user behavior trajectory features with respect to the user representation vector learned from the MIL module. The entire multi-scenario UIP pipeline is illustrated in Figure 2.

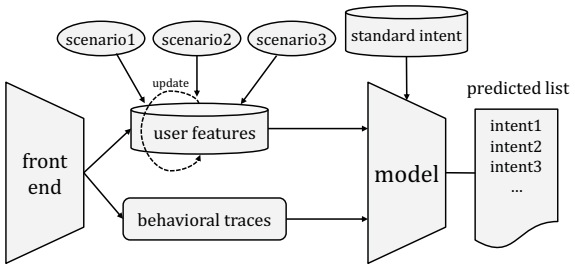

Figure 2: Illustration of the UIP pipeline for online service bots with various scenarios.

The main contributions of this paper are summarized below:

- We propose a framework for User Intent Prediction under the circumstance where valid user features drift with the evolution of related scenarios. The user features are learned using multiple instance learning methods, where a self-attention mechanism determines the instance weights through a deep neural network. To the best of our knowledge, it is the first work to marry MIL with the UIP task to address the feature drift problem.
- We also introduce a novel ratio-sensitive loss to handle the extreme class imbalances problem in online UIP scenarios. The loss weights for each sample is adjusted corresponding to the relative ratio of the prior probabilities between the predicted class and the ground truth class.
- We propose a sequential attention strategy that combines the LSTM module with an attention mechanism, in which the extracted user representation from MIL module are further utilized to capture critical user behaviors for better intent prediction.

2 RELATED WORK

2.1 User Intent Prediction (UIP)

UIP can be regarded as a task of recommending questions that users are interested in, which is similar to the task of CTR prediction that estimates the clicking likelihood of a user, given features of the user and candidate items. Usually, the input feature is extremely sparse and high-dimensional [13, 24]. As a pioneer work, NNLM [3] applied embedding technique to learn distributed representation for sparse feature id, which avoided the dimension explosion problem when dealing with large-scale sparse inputs. Traditional collaborative filter [18] embedded item and user features into a common representation space and estimated the probability of click by comparing the similarity of the mapped item-user pair. Meanwhile, there were works concentrating on feature intersection to capture the interrelationship between different fields. FM [23] modeled the first-order and second-order feature intersection explicitly, which demonstrated its effectiveness in many recommendation tasks. FFM [16] and AFM [30] aimed at modeling cross-field second order interaction, which is considered more effective than interaction within the field. Based on this paradigm, more and more models pay attention to the interaction between

features, including Wide&Deep [8], DeepFM [12] and DCN [29], where low-order and high-order features are combined to improve expression ability of the model. Subsequent works focus on higher order interaction through feed-forward network [13, 24, 31], attention mechanism [26] or feature-wise outer production [19]. To utilize user behavior sequence, recent studies [33, 34] introduce sequential learning structures to capture user interests from historical behaviors.

2.2 Multiple Instance Learning (MIL)

Multiple Instance Learning (MIL) [11] deals with the tasks in which each sample is represented by a bag of instances, and a label is attached only to the bag rather than each instance. It is also known as *learning from weakly annotated* [21]. MIL has been widely applied in many domains, such as drug activity prediction [11], web mining [35] and text classification [32]. Many approaches have been proposed for MIL through these years. They can be roughly classified into three categories, i.e., instance-space methods, bag-space methods and embedding-space methods. Among all MIL approaches, methods that extract global bag-level information, have been shown to achieve superior performance in general [1]. Recently, attention based method [15] is explored in MIL problems and promising results can be achieved.

3 PROBLEM STATEMENT AND NOTATIONS

In contrast to the conventional recommendation problems, in which each sample is described by both user and item features, UIP task deals with the sample that contains only the user-related features. The user features consist of two parts, i.e., the basic features and the behavior traces. The basic features encode the basic information of each user. The behavioral traces reflect the recent interests of the user. Each user is associated with a large number of labels, which denote whether the user is interested in a specific question or not.

Formally speaking, the features of each sample can be denoted as $[\boldsymbol{x}, \boldsymbol{s}]$, in which $\boldsymbol{x}$ is a vector that contains the basic features of this user, and $\boldsymbol{s}$ is a vector that contains the behavior traces of this user. The basic features can be represented as $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_M]$, where $\boldsymbol{x}_i$ represents a specific feature i (e.g., gender, occupation, etc.) in one-hot form. M denotes the total number of features. A N-step behavior traces vector is denoted as $\boldsymbol{s} = [s_1, s_2, ..., s_N]$, where $s_i \in R^{K_s}$ represents a single behavior id in one-hot form.

4 METHODOLOGY

In this section, we will present the proposed aDMSCN framework and its key components, i.e., attention-based MIL strategy and ratio-sensitive loss, which will be elaborated in detail.

4.1 Framework Overview

The whole architecture of the framework is illustrated in Figure 3. We separate the whole framework into several modules, which will be briefly described below.

4.1.1 Feature embedding. The embedding technique is used to obtain dense representation from original high-dimensional sparse feature. The embedding for one feature $\boldsymbol{x}_i$ can be obtained by $\boldsymbol{e}_i = E_i \boldsymbol{x}_i$, where $E_i \in R^{D_f \times (K_i)}$ is the embedding matrix, D_f is the feature embedding dimension All embeddings are stacked into one vector, i.e.

$$\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_M]. \tag{1}$$

Similarly, we embed the sequential feature with $\boldsymbol{e}_s = E_s \boldsymbol{s}$ to allow the interaction between basic features and behavioral traces, where $E_s \in R^{D_s \times K_s}$.

4.1.2 feature information aggregation. Based on the embedding of each basic feature, a final user representation can be obtained with proper operation. Commonly adopted strategy concatenates all feature embeddings, which would cause nontrivial noise with feature drift according to the discussion above. Instead, a permutation-invariant operation such as max pooling or average pooling is applied to extract stable aggregated representation. However, in practice, those operations cannot capture the most informative features which speculate the user's intent. In this paper, we propose to formalize this problem as a multiple instance problem and propose an attention-based method to learn the contribution weight of each feature, which will be presented in 4.2.

4.1.3 Behavior information aggregation. Besides user features, historical behavioral trajectory also contains rich information about the latent user intent. Considering that the length of behavioral trace varies among users, we take LSTM [14] for dynamic behavior feature representation.

The LSTM model focuses on all the historical behaviors with equal attention, while in most cases, only a few critical behaviors are sufficient to decide the current intent of the user. The other behaviors are just transitions between these critical ones. Locating key behaviors can help with predicting user intent and filtering out unrelated actions. Empirically, critical behaviors connected closely with the dynamic features. Therefore, we add attention mechanisms to the hidden state of LSTM and utilize the profile representation as the align vector to highlight the critical behaviors. The procedure is shown in Equation 2.

$$\begin{aligned} a_{s_i} &= \frac{\exp(\phi(\boldsymbol{h}_{s_i}, \boldsymbol{h}_f))}{\sum_{j=1}^{N} \exp(\phi(\boldsymbol{h}_{s_j}, \boldsymbol{h}_f))}, \\ \phi(\boldsymbol{h}_{s_i}, \boldsymbol{h}_f) &= \langle W_K \boldsymbol{h}_{s_i}, W_Q \boldsymbol{h}_f \rangle, \\ \boldsymbol{h}_s &= \sum_{i=1}^{N} a_i \boldsymbol{h}_{s_i}, \end{aligned} \tag{2}$$

where $\boldsymbol{h}_f$ denotes the feature representation from the MIL pooling model, $\phi(\cdot, \cdot)$ is the attention function, $W_K \in R^{D' \times D_s}$, $W_Q \in R^{D' \times D_f}$ are transformation matrices, D' is the dimension of attention space and $\boldsymbol{h}_s \in R^{D_s}$ is the final output for the attention-based LSTM.

4.1.4 Cross & MLP layer. Feature interaction is necessary to learn high-order feature combinations. We adopt the base architecture of DCN [29] for automatic bit-wise feature interaction. Different from the fully-connected feed forward layer, the cross layer is calculated by the following equation:

$$\boldsymbol{x}_k = \boldsymbol{x}_0 \boldsymbol{x}_{k-1}^T \boldsymbol{w}_k + \boldsymbol{b}_k + \boldsymbol{x}_{k-1}, \tag{3}$$

where w_k, b_k are parameters for k-th cross layer and x_k is the output of the k-th layer. Such cross operation could learn a high-order feature interaction, which approximates a special type of polynomial. Following the original architecture, we utilize fully

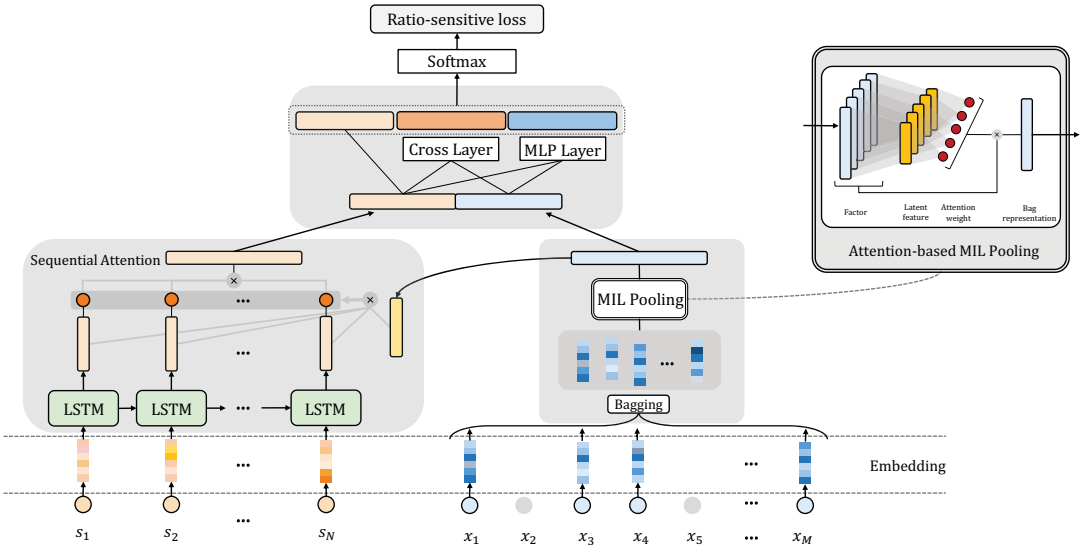

Figure 3: The overall aDMSCN framework. The sparse user features are first embedded into dense representations. Then, we bag the basic features into the MIL module to obtain a stable user profile representation (right branch). Historical behavior traces are processed by a sequential attention network, where profile representations are also associated as alignment vectors to highlight critical behaviors (left branch). Finally, we concatenate the information extracted from both behavior traces and basics features and fed into the Cross layers as well as the MLP layers for high-order feature integration, and a ratio-sensitive loss is adopted to deal with the extreme class imbalance problems.

connected layers for implicit feature interaction. The represented features from the two layers and the input behavior embedding are concatenated together at the output, as shown in Figure 3.

However, we need to address that, in the problem of UIP, two critical issues, i.e., *feature drift* [9] and *class imbalance*, will severely affect the performance of the whole algorithm. In this paper, pointed strategies are proposed to handle these problems. On the one hand, we formalize the UIP problem as multi-instance learning (MIL) task and handle it with an attention-based method, so that we can conquer the feature drift issue and capture critical features as the same time. On the other hand, a ratio-sensitive loss is proposed, to alleviated the negative effect of class imbalance. We will explain these two parts in details as below.

4.2 Attention-based MIL Strategy

As mentioned above, some features may become or cease to be relevant to the prediction task with the evolution of the scenarios. However, existing CTR prediction models usually require stable features during the training and inferring process and therefore have little adaptability to such feature drift. For example, let $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_4, \boldsymbol{e}_5]$ represent embedding for feature $\{A, B, C, D, E\}$ respectively. As business changes, $\boldsymbol{e}_3, \boldsymbol{e}_4, \boldsymbol{e}_5$ may become very important while $\boldsymbol{e}_1, \boldsymbol{e}_2$ become irrelevant, causing the drift of features. Instead, we consider the feature embedding set and each feature embedding as the instance bag $\mathcal{B}$ and instance $\boldsymbol{x}$, respectively. Then an embedding-space method is developed to extract bag representation from instance features by

$$S(\mathcal{B}) = g(\sum_{\boldsymbol{x} \in \mathcal{B}} f(\boldsymbol{x})), \tag{4}$$

where $S(\mathcal{B})$ is a symmetric function that is permutation-invariant to the instances. Notably, the simple strategy average pooling for feature drift, i.e.,

$$\boldsymbol{e}_{avg} = \frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{x}_i \in \mathcal{B}} \boldsymbol{e}_i, \tag{5}$$

can be regarded as a vanilla MIL approach, where $\mathcal{B}$ denotes the valid feature set and $\boldsymbol{e}_i$ is the corresponding embedding for feature $\boldsymbol{x}_i$.

Since different features may contribute dramatically different for the final predicted result, the average pooling method may not be a good choice. In this work, we proposed to employ the attention mechanism to learn the contribution weight of each feature. Concretely, the corresponding weight a_i can be calculated as below,

$$\begin{aligned} \boldsymbol{h}_i &= \tanh(W_l^T \boldsymbol{e}_i), \\ m_i &= \psi(\boldsymbol{v} \oplus \boldsymbol{h}_i), \\ a_i &= \frac{e^{m_i}}{\sum_{j=1}^{N} e^{m_j}}, \end{aligned} \tag{6}$$

where $W_l \in R^{D_f \times D_l}, \boldsymbol{v} \in R^{D_l}$ are trainable parameters of the model, D_f, D_l is the dimension of the feature embedding space and the latent space respectively, ψ denotes a scoring function, which is a one-hidden-layer ReLU network to determine the compatibility scores m_i between vector $\boldsymbol{v}$ and $\boldsymbol{h}_i$. The attention weight a_i is calculated by softmaxing the compatibility score m_i and could reflect the consistency between the factor embedding $\boldsymbol{e}_i$ and the latent bag embedding. The ψ is equivalent to an attention function that decides the similarity between the query $\boldsymbol{v}$ and key $\boldsymbol{h}_i$, Here $\boldsymbol{v}$ could be regarded as the latent embedding vector of a pseudo

bag representation which is learned along with the model weights. The actual bag representation (i.e., the user feature) is calculated through weighted averaging of all feature embeddings $\boldsymbol{e}_i$:

$$\boldsymbol{h}_f = \sum_{\boldsymbol{x}_i \in \mathcal{B}} a_i \boldsymbol{e}_i. \tag{7}$$

The procedure of attention-based MIL is illustrated in the right part of Figure 3.

4.3 Ratio-sensitive Loss

In this paper, we treat UIP as a multi-classification problem, and choose negative log-likelihood function as the objective function. The classical cross-entropy function suffers from the imbalanced distribution of classes.The model might be overwhelmed by the major classes, leading to an increasing bias of the predicting results. A common strategy for the imbalance distribution introduces weighted cross-entropy is shown in Equation 8,

$$Q_w = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{y} \alpha_i y_i \log p_i(\boldsymbol{x}_j). \tag{8}$$

In practice, α_i is tuned to assign larger weight to samples of minor class. There was also work attempting to differentiate between easy and hard examples [20] through the predicting confidence of the model, as shown in Equation 9.

$$Q_h = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{y} \left(1 - p_i(\boldsymbol{x}_j)\right)^{\gamma} y_i \log p_i(\boldsymbol{x}_j), \tag{9}$$

where γ is a weight parameter.

Existing works model the class imbalance with respect to the ground truth or predicted result alone and have achieved promising results for binary classification tasks. However, for multi-classification with multiple major and minor classes presented, the relative balance between the predicted and ground truth class is neglected by the previous works. We argue that the proportion of both predicted and ground truth class should be considered when deciding loss weights. Concretely, more attention should be paid on those misclassifications from minor classes to major classes rather than the other way around. In this paper, we propose a novel ratio-sensitive loss to model the relative imbalance. The ratio-sensitive loss assigns distinct weight for each sample with respect to the class proportion ratio between the ground truth and the predicted class. The formulation is shown in Equation 10:

$$Q_r = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{y} r\left(y_i, \boldsymbol{p}\right) y_i \log p_i(\boldsymbol{x}_j), \tag{10}$$

where

$$r(y_i, \boldsymbol{p}) = \underbrace{\left(1 - \mathbf{p}(y_i)\right)}_{T_w} \underbrace{\sum_{j=1}^{y} p_j y_i \log_{\mathbf{p}(y_j)} \mathbf{p}(y_i)}_{T_r}. \tag{11}$$

It is composed of two components. The T_w term denotes the loss weights assignment with respect to the prior probability of the ground truth class, which is similar to the vanilla weighted loss. $\mathbf{p}(y_i)$ denotes the prior probability distribution for the i-th class. T_r denotes the ratio-sensitive term, where $\log_{\mathbf{p}(y_j)} \mathbf{p}(y_i)$ depicts the relative balance ratio between the predicted classes (the j-th class) and the ground truth class (the i-th class). Generally, the distribution ratios between classes vary significantly, causing extreme loss weight value. Thus, we use logarithms to soften the ratio to prevent gradient explosion.

Note that in most cases the class prior probability information is unknown during training. In these cases, we approximate $\mathbf{p}(y_i)$ with $\mathbf{p}(y_i) = \frac{N_{y_i}}{N}$, which is calculated by dynamically accumulating the respective amount within each training batch. The same accumulated prior distribution value is utilized during the inferring phase.

Table 1: Statistics of feature utilized by online user intent prediction module. Features are composed of sparse vector grouped in various fields.

feature type	category	dimension
4*user feature	gender	2
	occupation	~100
	...	...
	total	$\sim 10^7$
behavioral trace		$\sim 10^4$

5 EXPERIMENTAL STUDY

5.1 Dataset

To demonstrate the effectiveness of our framework for online UIP task, we conduct experiments on the customer service bot dataset of Alipay, the biggest mobile payment platform in China. This dataset contains one million records, which was collected from click logs with the time span of a month. Each record is tagged with a standard question representing a unique user intent interested by the customer. In total, there are approximately 7000 standard questions, 600 feature categories and 4000 behavior ids involved in the dataset. Each click log records the user id, timestamps, user features, clicked question's id and latest 300 historical behaviors of the user. The feature types involved are illustrated in Table 1. We sort the logs by date and select the logs recorded during the last day as the test set, which contains about 30,000 samples. The remaining logs are used for the training procedure.

5.2 Competitors

In this work we choose following approaches for baselines.

- **LR** Before the introduction of deep learning techniques, Logistic Regression (LR) is a widely used model for the CTR prediction task. It is listed here as a weak baseline.
- **MLP** Embedding followed by fully connected layers is adopted as the base architecture for many deep-based CTR modeling.
- **DCN** DCN is the base architecture of our framework and is used here as a strong baseline.
- **DeepFM** DeepFM [12] utilizes factorization machine for the crossing procedure. The remained part is similar to DCN.
- **xDeepFM** xDeepFM [19] focuses on explicit feature-level feature interaction through a novel Compressed Interaction

Network (CIN). It achieves promising results for CTR tasks when the categories of user features are stable.

- **AutoInt** AutoInt [26] performs automatic feature interaction through self-aligned weighted summation, which can be classified as a kind of self-attention mechanism utilized by Transformer [28].

5.3 Evaluation Protocols

To evaluate the performance of our model on UIP task, we introduce three popular metrics.

- **Hit Rate @ k** The Hit Rate (HR) is defined as $HR = \frac{N_{hit}}{N}$, where N is the total number of instances and N_{hit} is the number of instances in which the clicked question is presented in the top-k list.
- **Mean Average Precision** Mean Average Precision (MAP) is calculated by averaging the Average Precision (AP) for all classes.
- **Mean Reciprocal Rank @ k** Mean Reciprocal Rank (MRR) is the average of reciprocal ranks for the correctly-recommended items. It is defined as $MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$, where $rank_i$ is the rank position of the ground truth class of the i-th sample. When the rank exceeds top-k, the reciprocal rank is set to zero. A larger MRR indicates a more suitable ranking for the top-k item list.

5.4 Implementation Details

The embedding dimensions for DCN and aDMSCN were set in terms of the empirical formulation $D = 6\sqrt[4]{D_N}$, where D_N is the dimension of the sparse feature space. All models were trained through mini-batch stochastic with Adam [17] optimizer. The batch size was set at 1024 for AutoInt and 256 for others for the best performance. In DCN, we stacked three interaction layers for the crossing part and three fully connected layers for the dense part, with $1024, 512, 512$ as the size of hidden units respectively. Two interaction layers were stacked for the Compressed Interaction Network in xDeepFM. For AutoInt, we used an embedding size of 8 for the attention vector and stacked three interaction layers with two attention heads for each layer. Dropout was added to the output of each model. The gradient clip was set as 5. For xDeepFM and AutoInt model, all basic features and historical behaviors embeddings were concatenated and padded to a fixed length as input feature. For other baseline models that perform bit-wise interaction, the basic feature and behavior embeddings were averaging summed up separately and concatenated as input features. The parameters used by all baseline methods are tuned on the same training and development set to achieve the best performance. We applied exponential decay in the training procedure. The base learning rate was set to 0.01 and the decay rate was set to 0.9. For all experiments, an early stopping technique was applied at training step $350,000$ to prevent the model from overfitting.

5.5 Comparison with Baseline Methods

The performances of different methods are summarized in Table 2, with the best results highlighted in boldface. Table 2 indicates three key observations about the performance comparison among these approaches.

Table 2: Performance comparison on the offline dataset.

	HR@1	HR@6	MAP	MRR@6
LR	17.38%	34.33%	25.05%	0.2320
MLP	17.62%	34.66%	25.28%	0.2339
DCN	19.70%	37.75%	27.11%	0.2521
DeepFM	17.05%	32.68%	24.24%	0.2239
xDeepFM	14.38%	27.26%	20.40%	0.1867
AutoInt	18.36%	36.12%	27.25%	0.2544
aDMSCN(ours)	**20.94%**	**39.61%**	**29.11%**	**0.2725**

- Little performance improvement is achieved through stacking additional fully-connected layers based on the comparison between LR and MLP. With DCN, however, a significant improvement can be observed by introducing interaction layers, which indicates the importance of effective feature interaction.
- The deep models that involve feature-level interaction present consistent performance downgrades compared to their promising results on several recommendation benchmarks. Particularly, xDeepFM achieved the worst performance. These results contradict the results of some previous works using other benchmarks [19, 26]. We ascribe such contradiction to the feature drift issue existing in our UIP task. Notably, the noise from feature drift is magnified with the increase of the order of feature interaction. Thus, the higher the order of feature-wise interaction, the worse performance the model would achieve. High-order interaction would, therefore, degenerate the model performance. Meanwhile, aDMSCN can overcome the influence of feature drift noise, as it integrates features and only deals with element-wise feature interaction.
- Our proposed framework aDMSCN outperforms all other baseline models. Compared with the best baseline method, our aDMSCN makes a relative performance improvement by up to 6.29%. Notably, in recommendation-related tasks, a 0.1%-level improvement is significant [5, 34], especially for practical industrial application scenarios with enormous daily active users and page views. aDMSCN presents promising superiority when dealing with online UIP tasks. The proposed models provide effective help to free aDMSCN from the issues brought by multiple scenarios in online service.

5.6 Ablation Study

In order to analyze the benefits brought by the three models introduced in our framework, we perform an ablation study that integrates the proposed models separately with the basic model Deep Sequential Cross Network (DSCN), i.e., DCN with LSTM for behavioral trace and average pooling for basic feature. The results are shown in Table 3, in which *SA* denotes *Sequential Attention*, *RS* denotes *Ratio-Sensitive loss* and *MIL* denotes *attention-based MIL pooling*. Table 3 demonstrates that all three models contribute to the

Table 3: Ablation study for various models in our framework.

	HR@1	HR@6	MAP	MRR@6
DSCN	20.17%	38.73%	24.84%	0.2653
DSCN&SA	20.31%	38.78%	27.21%	0.2656
DSCN&RS	20.42%	38.83%	28.55%	0.2657
DSCN&MIL	20.89%	39.55%	29.08%	0.2720

Table 4: Comparison for various methods dealing with class imbalance problem.

	HR@1	HR@6	MAP	MRR@6
Vanilla cross entropy	20.17%	38.73%	28.24%	0.2656
Label normalization	20.09%	38.38%	28.02%	0.2583
Vanilla weighted loss	20.12%	38.39%	28.16%	0.2624
Focal loss	20.33%	38.67%	28.38%	0.2654
Ratio-sensitive loss	20.42%	38.83%	28.55%	0.2657

performance improvement of our framework. Among these models, the MIL module contributes to the most significant improvement. It is consistent with the empirical observation for online service, where the feature drift issue limits the performance of most existing models.

5.7 Evaluation of the Ratio-sensitive Loss

Currently, approaches for class imbalance can be broadly divided into sampling-based methods and weighted-loss-based methods. The sampling-base methods, e.g., SMOTE [4], require measurable feature and is infeasible for sparse features used in recommendation and UIP tasks. Therefore, in this work, we mainly focus on weighted-loss-based methods comparison. Label normalization [27], vanilla weighted loss and focal loss [20] are selected as baselines. These methods are combined with the base model DSCN separately. The results in Table 4 show that label normalization and vanilla weighted loss make little contribution to the model performance. Meanwhile, the ratio-sensitive loss outperforms other methods and achieves a 1.2% relative gain in terms of HR@1. Even compared with Focal loss, the ratio-sensitive loss still achieves a 0.1% absolute HR@1 gain as it takes both prediction and ground-truth information into consideration when deciding loss weights.

We also visualize the statistical class distribution of the model output and compare it with the prior class distribution of dataset. Figure 4 shows that the model with classical cross-entropy loss suffers from imbalanced data. The majority class overwhelms information of minor classes in the training process, which may lead to poor distribution prediction for minor classes. Therefore, significant bias can be observed in the statistical distribution of the predicted class. In contrast to the classical cross-entropy loss, the ratio-sensitive loss takes into consideration the relative balance/imbalance between prediction and ground truth. Therefore the class distribution learned by the model using ratio-sensitive loss presents more consistency with the prior distribution. Figure 4 illustrates the significant improvement of class distribution prediction for imbalanced classes with our proposed ratio-sensitive loss

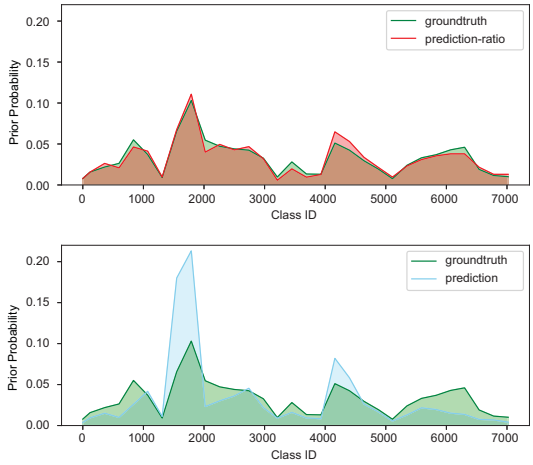

Figure 4: The comparison between the prior distribution (ground truth) and the class distributions predicted by the model with the ratio-sensitive loss (upper sub-figure) and classical cross-entropy loss (lower sub-figure), respectively.

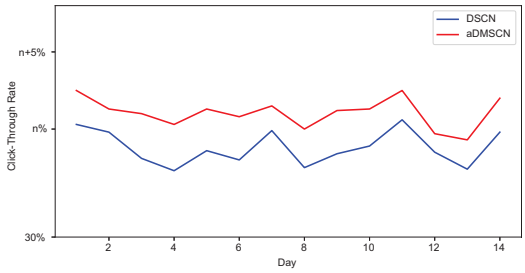

Figure 5: Online A/B testing comparison between baseline model DSCN and aDMSCN. Instead of the specific CTR scores, we use n and $n+5$ as indicators due to confidentiality reasons.

function and demonstrates the potential of our loss function for multiple imbalanced classes.

5.8 Online Evaluation

We deployed our model online in an E-payments service bot and conducted a 14 days A/B testing. During the testing, our aDMSCN framework is compared with the basic DSCN model using daily CTR as the criteria. While the specific CTR scores can not be disclosed due to confidentiality reasons, we use an indicator n to demonstrate the superiority of the proposed methods. The results are shown in Figure 5. Despite some slight fluctuation, aDMSCN outperformed DSCN during the whole testing phase. On average, the proposed framework brings a relative improvement of 5.36% in terms of CTR and 4.85% in terms of user satisfaction rate, which is considered as a significant improvement for the platform. Note that there were around 30 features from two business scenarios becoming invalid during this two-week testing as a result of feature drift. It indicates that the proposed framework can alleviate the impact of feature drift

and therefore more suitable for practical scenarios where feature drift constantly occurs.

6 CONCLUSION

In this paper, we propose a UIP framework to handle multiple business scenarios evolution in practical online service platform. We treat the user sample as a bag with all valid features as instances, formulating the feature drift issue with the MIL model. An attention-based method is proposed to extract user representation from the feature embeddings and highlight the key features simultaneously. A novel ratio-sensitive loss is proposed to learn adaptive weighted loss, in terms of the relative ratio of the prior probability between the ground truth and the predicted class. Besides, we combine the attention mechanism with the LSTM model to capture critical user behaviors for better behavioral trace representations. Experiments on real-world datasets demonstrate that our framework achieves better performance for UIP task compared with alternative state-of-the-art methods.

REFERENCES

[1] Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence* 201 (2013), 81–105.

[2] Jean Paul Barddal, Heitor Murilo Gomes, Fabrício Enembreck, and Bernhard Pfahringer. [n.d.]. A survey on feature drift adaptation: Definition, benchmark, challenges and future directions. *Journal of Systems Software* ([n. d.]), S0164121216301030.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[5] Cen Chen, Chilin Fu, Xu Hu, Xiaolu Zhang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. 2019. Reinforcement Learning for User Intent Prediction in Customer Service Bots. In *Proceedings of the 42Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. ACM, New York, NY, USA, 1265–1268. https://doi.org/10.1145/3331184.3331370

[6] Cen Chen, Xiaolu Zhang, Sheng Ju, Chilin Fu, Caizhi Tang, Jun Zhou, and Xiaolong Li. 2019. AntProphet: an Intention Mining System behind Alipay's Intelligent Customer Service Bot. 6497–6499. https://www.ijcai.org/proceedings/2019/935

[7] Cen Chen, Peilin Zhao, Longfei Li, Jun Zhou, Xiaolong Li, and Minghui Qiu. 2017. Locally Connected Deep Learning Framework for Industrial-scale Recommender Systems. In *WWW*.

[8] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, and Mustafa Ispir. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10. https://doi.org/10/gfwc7n

[9] Ofer Dekel, Ohad Shamir, and Lin Xiao. 2010. Learning to classify with missing and corrupted features. *Machine learning* 81, 2 (2010), 149–178.

[10] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.

[11] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89, 1-2 (1997), 31–71. https://doi.org/10/b738d7

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Melbourne, Australia, 1725–1731. https://doi.org/10.24963/ijcai.2017/239

[13] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364. https://doi.org/10/gfvsvd

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780. https://doi.org/10/bxd65w

[15] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*.

[16] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 43–50. https://doi.org/10/gf5psn

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[18] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 447–456. https://doi.org/10/dhw5qt

[19] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1754–1763. https://doi.org/10/gf3nkw

[20] 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. https://doi.org/10/gf226d

[21] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. 2014. Weakly supervised object recognition with convolutional neural networks. In *Proc. of NIPS*.

[22] Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. ACM, 25–33.

[23] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000. https://doi.org/10/bgfknx

[24] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and J. C. Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 255–262.

[25] Jie Shen, Ying Gao, Cang Chen, and HaiPing Gong. 2012. A rank-based Prediction Algorithm of Learning User's Intention. *Physics Procedia* 24 (2012), 1742–1748.

[26] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *CIKM*.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. ACM, 12. https://doi.org/10/gf2wst

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

[31] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*. Springer, 45–57. https://doi.org/10/gf5psm

[32] Ya-Lin Zhang and Zhi-Hua Zhou. 2017. Multi-instance Learning with Key Instance Shift. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 3441–3447. http://dl.acm.org/citation.cfm?id=3172077.3172370

[33] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. [n.d.]. Deep Interest Evolution Network for Click-Through Rate Prediction.

[34] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, USA, 1059–1068. https://doi.org/10/gfx98v

[35] Zhi-Hua Zhou, Kai Jiang, and Ming Li. 2005. Multi-instance learning based web mining. *Applied Intelligence* 22, 2 (2005), 135–147.