NAUI: Neural Attentive User Interest Model for Cross-Domain CTR Prediction

Ren Wang1,a, Zhihong Xie2, Gaofeng Qi2, and Ping Li*

1Changsha University of Science & Technology of Computer and Communication Engineering, Changsha 410114, Hunan, China
2Changsha University of Science & Technology of Computer and Communication Engineering, Changsha 410114, Hunan, China
aemail: 775626063@qq.com
*email: lping9188@csust.edu.cn

Abstract. In the existing works for recommender systems, the design of Cross-Domain CTR prediction is less mentioned. In order to solve the data sparsity problem in CTR prediction, Cross-Domain Recommendation (CDR) leverages a wealth of information from a source domain to improve the CTR prediction performance on a target domain with sparse information, which is better than Single-Domain CTR prediction. Ads are usually displayed with natural content, which gives Cross-Domain CTR prediction the opportunity for problem-solving. In this paper, we propose a CDR novel named NAUI which can leverage auxiliary data information to improve the CTR prediction performance. NAUI utilizes an extracting user interest method which reduces excessive punishment for active users, and jointly training Cross Net and MLP components together on CDR. NAUI can capture more information about nonlinear features and combined features efficiently capture implicit and explicit high-order features interactions, and greatly improve the expression ability of the model on Cross-Domain CTR recommendation. Furthermore, we add an auxiliary classifier to the deep neural network to improve recommendation performance. Our experiment results has demonstrated that NAUI outperforms several frequently state-of-the-art methods of CTR prediction.

1. Introduction
Click-Through Rate (CTR) prediction is a critical task in the online advertising industry and recommender systems. Accurately predicting the click-through rate (CTR) is to predict the probability that people will click on particular ads or an item. As an example, the ad ranking strategy is generally designed for \( CTR \times bid \), where the bid is the profit the ad platform receives if an ad is clicked by a people. Therefore, the performance of the CTR prediction model has a direct impact on the ad platform’s revenue, it is crucial to accurately estimate the CTR of ads.

Modeling CTR prediction has attracted considerable attention to both academia and the industry community in the past few years [1, 4, 5, 9]. Many scholars have put forward some models for exploration and analysis in this field such as Logistic Regression (LR) [18], and factorization machines (FMS) based models [7]. Recently, with the great development of deep learning technology in many fields which can reduce a great many of feature engineering jobs such as computer vision (CV) [29] and speech recognition [30] and natural language processing (NLP) [31], many deep
learning-based methods have been proposed for CTR prediction [9, 28, 4, 26]. Therefore, some neural network features learning-based models now have become popular with CTR prediction tasks. For example, Factorization-Machine Supported Neural Networks (FNN) [9], Neural factorization machines (NFM) [28], Attentional Factorization Machines (AFM) [17], Wide&Deep (WDL) [1], Deep&cross (DCN) [4], Deep Interest Network (DIN) [5], DeepFM [2], XDeepFM [19], DSTN [26].

As above mentions, recent exiting works recently mainly focus on single-domain CTR prediction. These methods only exploit single-domain data information about CTR prediction, and these exiting methods model aspects such as feature interaction [7], contextual information [5], and user behavior history [26, 5]. Moreover, Current recommendation systems (RSs) based on single-domain CTR prediction have a static problem of data sparseness. A relatively effective method of problem-solving is to transfer relatively richer data information about the source domain based on overlapping users, to improve the recommendation accuracy on the target domain with relatively sparse data information, which is called Cross-Domain Recommendation (CDR). Ads are usually presented as natural content, which provides a powerful condition for Cross-Domain CTR prediction.

The main goal of cross-domain CTR recommendation is preference mapping between the two relevant-domains, that the major advantage is to richer data information based on cross domains, which can alleviate the problem of data sparsity effectively and cold-start problem to improve the quality of CTR prediction. In order to make the user preference mapping possible, existing approaches such as EMCDR [12] encodes features of single vectors. And they conduct cross-domain mapping as a whole. The existing solutions to learn user/item representations of the both domains respectively, as shown in Figure 1. Then, according to the overlapping users of two domains, the cross-domain preference representation mapping is learned to realize information transfer.

In our study, we assume that the user’s interest is mixed which contains static inherent interests and attentive user interests in a certain period of time. Modeling interest areas and exploring their interrelationships between different fields will promote a better understanding of user preferences and offer effective support. Our study divides the two data onto the source news domain and the ads domain, user preference expression has corresponding user behavior based on the overlapping users between the domains, for instance, the news that the user just viewed or the ad that the user has recently clicked. A piece of news may exist some correlation between the target ad (the contents of the two may be different). For example, a user is likely to click a Game advertisement after viewing some Entertainment news. What advertisements the active user has recently clicked may have some positive effects on what advertisement the user may click in the short future. Each user profile feature such as user id, age group, city, and gender, reflects a user’s static interest. We can transfer some useful
information from the source domain to the target domain by analyzing the relationship between these factors.

In the cross-domain recommendation scenario, we will discuss how to learn user preferences to improve CTR performance. We examine some previous work and propose the Neural Attentive User Interest Model (NAUI), whose structure is given in Figure 2. In NAUI, models two layers: interest transfer layer, which aim to extract user interest include the static user intrinsic interest $p_u$ based on cross-domain and the attentive interest $a_s/t$, $s,t \in D_s,D_t$; interaction & prediction learning layer, which by adding MLP and Cross network to exploit efficient learning high-order feature interaction, and by adding auxiliary classifiers under each middle layer of MLP to simplify the training process.

The main contributions to this paper are summarized as follows:

By using an attention mechanism, the NAUI can distill easily more information about user interest by smoothing user history and adjusting attention mechanism pressure between both domains, and adaptively obtain attentive user interest in recently clicked news/ads to transfer related knowledge.

Using both MLP and Cross network to exploit efficient learning high-order feature interaction, can capture more information about nonlinear features and combined features, and greatly improve the expressive ability of the model for Cross-Domain CTR recommendation.

Inspired by NON [20] and GoogLeNet [16] by adding auxiliary classifiers under each middle layer of MLP to simplify the training process and improve the discriminative power of the model.

We conduct experiments and detailed analysis on both public and UC Toutiao real-world datasets. Offline experimental results NAUI outperforms state-of-the-art methods for more accurate CTR prediction.

2. Related Work

Our work is related to two main sub-area categories of recommender systems (RSs): Single-Domain CTR recommendation, and Cross-domain CTR recommendation.

2.1 Single-Domain CTR Recommendation

Modeling CTR prediction has attracted lots of attention from both academia and industry [5] [11] [9] [14] [37]. Existing work mainly addresses the single-domain CTR prediction recommendation. Although generalized linear model Logistic Regression (LR) [18] has shown decent performance in practice, it hard to learn sophisticated feature interactions [3]. Traditional Factorization Machines (FMs) [7] use the dot product of two vectors to model pairwise feature interactions to address this limitation. Field-weighted Factorization Machines [27] and Field-aware Factorization Machines (FFMs) [8] extended the ideas of FMs further consider leveraging the effect of the field information to improve the performance of FMs model.

Many deep learning based CTR models have been proposed such as deep neural networks (DNN) and product-based neural network (PNN) is exploited for Single-Domain CTR prediction and user/item recommendation in recent years so that automatically learn feature representations and high-order feature interactions [2, 9, 25]. Some methods such as Wide & Deep Learning [1], DeepFM [2], NFM [10], xDeepFM [19], and so on, combines a shallow part and a deep part in order to capture more informative feature interactions (low- and high-order). Deep Interest Network (DIN) [5] and its upgraded DIEN [13] model user interest based on historical click behavior. Yin et al. [23] and Xiong et al. [24] enhance recommendation consider varying the effect of contextual factors such as ad interaction, add depth, and query diversity. DSTN [26] jointly exploits contextual information, click information (has clicked and unclicked) for CTR recommendation.

2.2 Cross-Domain CTR Recommendation

Cross-domain CTR recommendation being aimed at improving the recommendation effect of the target domain by leveraging relevant source domain as auxiliary information, so that relief the problem of data sparsity and cold-start in the recommendation system. Some Collaborative recommendation
approaches utilize interaction information (e.g., clicks, ratings) across domains to deal with the data sparsity problem. For instance, Collective Matrix Factorization (CMF) [15] method concatenation multiple rating matrices, then sharing user factors across domains to achieve knowledge integration across domains. To avoid the leak of user privacy, NATR [8] is not sharing user relevant data for a cross-domain recommendation. CoNet [6] realize the dual knowledge transfer across domains by introducing cross-connections units to improve CTR performance. Content-based existing work creates links by the common user/item attributes. For example, Elkahky et al. [14] transform knowledge information based on the latent space matching between user intrinsic profiles and item descriptions. CKE [22] proposes utilizing structure, textual, and visual knowledge of items as auxiliary knowledge information to offer assistance in the learning of feature representation. Hybrid learning models combine interaction data information and attribute data to enhance the model quality. For example, CCCFNet [10] combines collaborative filtering methods and content-based filtering approaches in a unified framework. Numerous deep learning-based methods are applied to enhance knowledge transfer in Cross-domain recommendations. EMCDR [12] tries to use the Multi-Layer Perceptron (MLP) and a transformation matrix to map the feature vector across domains. MiNet [11] learning user interest by an attention mechanism and through a multi-layer fully connected neural network.

In our work, we propose a CDR framework to address the CTR prediction problem. First, we model user intrinsic profiles as static interest and attentive user interest. Next, fuse them adaptively in a neural network learning framework with the classifier.

3. The NAUI Framework

Figure 2 depicts an overview of the model architecture. In this section, we elaborate on the Neural Attentive User Interest (NAUI) model for cross-domain CTR prediction. We first introduce the problem statement and preliminaries. In the interest transfer layer, we propose a smoothing method to learn and fusion user interest in the attention technology. Next in the interaction & prediction learning layer, and the Deep & Cross Network (DCN) to learn feature interactions on Cross-Domain Recommendation. Finally, we present a classifier optimizer as an auxiliary loss to efficiently train the model in an unsupervised manner.

Notations. We represent vectors, matrices, and scalars as bold capital letters (e.g., X), bold lowercase letters (e.g., x), and normal lowercase letters (e.g., x) in NAUI, respectively. Combined with the representation of the domain, then:(1) $D_s$ represents the source domain, and (2) $D_t$ represents the target domain. $D_u$ represents the common user in both domains. Without explanation, all feature vectors are located in a column form.
3.1 CTR Problem Statement

The fundamental task of CTR prediction in advertising for the company is to build a prediction model to predict the probability that people will click on a specific ad. Every instance in the real-world dataset can be described by multi-field such as user aspect (User ID, City, etc.), and ad aspect (Creative ID, Title, etc.), and so on.

Often, we can regard the CTR problem about cross domains as leveraging the data information for user preference mapping to improve the prediction performance from a source domain (or source domains) to a target domain. Without loss of generality for the news feed advertising, the source domain \( D_s \) and the target domain \( D_t \) denote the natural news feed and the advertisement respectively. In this case, a certain degree of overlap (e.g., common users) between the users of different domains is necessary, which can be used to connect the two domains and share knowledge information across them, but there are no overlapped items. Differently, in NAUI, we explore auxiliary classifiers for improving the CTR prediction.

3.2 Preliminaries

**Embedding Layer.** An embedding layer is applied upon the original binary feature (high-dimensional, sparse) input to compress it into a low-dimensional and dense real-value vector. We first encode features into one-hot vectors, e.g. \([0,1,0]\)”. For the i-th vector, its one-hot vector is \( \psi_i = \text{one-hot}(i) \), where \( \psi_i \in \mathbb{R}^N \) is a vector, \( N \) is the number of unique features. The set of unique feature vectors covers all the features in both domains, would to enable Cross-Domain knowledge information transfer. Note that we define an embedding matrix \( E \in \mathbb{R}^{D \times N} \) ( \( D \ll N \)). Vector \( e_i \in \mathbb{R}^D \) denotes the corresponding embedding i-th features, and \( e_i = Ev_i \). In general, \( e^s \in D_s \) denotes the source domain vectors, \( e^t \in D_t \) denotes the target domain vectors.

**Multiple Layer Perceptron (MLP).** Given the concatenation dense representation vector, FNN, Deep Crossing(DCN), and the deep part in Wide & Deep(WDL) exploit a MLP network on the embedding feature vector \( e \) to learn feature interactions(high-order) automatically.

**Cross Net.** Inspire by [4] propose the Cross Network (Cross Net), whose architecture is shown in Figure 3, to apply explicit feature crossing. Its purposes are to explicitly model learn the feature interactions (high-order). Cross Net is composed of special cross layers, and Cross Net in every layer as follows the formula:
\[ z^{L+1} = z^0 (z^L)^T w^L + b^L + z' = f(z^L, w^L, b^L) + z^L . \tag{1} \]

where \( z^L, z^{L+1} \in \mathbb{R}^d \) features vectors(column format) represents the outputs from the \( L \)-th and \( (L+1) \)-th special cross layers, respectively; \( w^L, b^L \in \mathbb{R}^d \) are the weight and bias parameters of the \( L \)-th Cross Net layer. Special cross per-layer adds back its input after a feature crossing \( f \), where the mapping function \( f : \mathbb{R}^d \to \mathbb{R}^d \) fits the residual of \( z^{L+1} - z^L \).

**Loss Function.** For base CTR prediction model training, the objective function used is the negative log-likelihood function defined as:

\[ \ell = -\frac{1}{|Y|} \sum_{y \in Y} [y \log \hat{y} + (1 - y) \log(1 - \hat{y})], \tag{2} \]

where \( y \in \{0,1\} \) as the label, represents whether the associated user clicked the instance corresponding to the prediction CTR value \( \hat{y} \). \( Y \) denotes the collection of labels in real-data. In principle, we have two loss function loss \( \ell_t \in D_t \& \ell_s \in D_s \) in both domains.

### 3.3 Interest transfer layer

We put each ad instance features split two features: user features and ad features. And put ad feature embedding vectors to concatenate into an ad representation vector \( r^t \in \mathbb{R}^{D_t} \) in the target domain. Similarly, we can obtain a new representation vector \( r^s \in \mathbb{R}^{D_s} \) about the source domain. We are inclined to concatenate because concatenation can preserve the complete feature representation of both domains, rather than feature pooling technology. For a target user \( u \), her static interest representation vector \( r_u \in \mathbb{R}^{D_u} \) is obtained by concatenation the corresponding user embedding. In general, the user static interest representation vector is represented by user inherent profiles.

Let \( \{r^{3}_{i}\}_{i} \) denote the set of representation embedding vectors of news(has recently clicked), and the set of representation embedding vectors of ads (has recently clicked) as \( \{r^{f}_{j}\} \). A target user is under a high probability to click an ad after viewing some random items (news/ads). There may be certain existing correlation information between the viewed news and the target ad. Even if the contents of both are completely different. Hence we can effectively transfer some useful information from the source domain to the target domain base on using such relationships. From time to time, due to the number of items (has clicked news/ads) may be different, we need to aggregate these items (has clicked news/ads) as follows

\[ a_s = \sum_i \alpha_i r^s_i, a_t = \sum_j \beta_j r^t_j, \tag{3} \]

where \( \alpha_i, \beta_j \) is a weight value assigned to \( r^s_i, r^t_j \) (indicate items importance). The aggregated scores \( a_{st} \) reflect the interest of users from both domains.

**Smooth Attentive User Interest.** However, it’s an unreasonable treatment in a practice that every item(has clicked news/ads) are regarded as equal importance, the principal reason is that some existing news may not be indicative effect for the target ad (each of \( \alpha_i, \beta_j \)should be different). Attention Net
Mechanism model offers us a better method to calculate the weight. In conventional usage scenarios of attention mechanism such as computer vision (CV) and natural language processing (NLP) tasks, the number of attentive components does not vary much (e.g., regions in images, words in sentences, and so on). That is the reason in the right way and then, has a great probabilistic explanation weight. However, such a situation does not exist anymore for user historical data records as a result of the historical log length of users (i.e., the number of historical log items clicked by target users) that can vary much. We found the softmax function performs $\eta$, normalization on attention weight value may have an overly punish the weight value of active users with a long history. However, how to use it is still flexible. A specific attention method is to compute weights $\alpha_s, \beta_t$ as

$$
\alpha_i = \frac{\text{exp} (\vec{\gamma}_i)}{\sum \text{exp} (\vec{\gamma}_j)}, \quad \beta_j = \frac{\text{exp} (\vec{\gamma}_j)}{\sum \text{exp} (\vec{\gamma}_j)} (4)
$$

where $\eta$ is the interest smoothing exponent variable, a set hyper-parameter is in the range of $[0, 1]$. Obviously, when $\eta = 1$, it recovers the softmax function; when $0 < \eta < 1$, the value of the function denominator will be suppressed, as a result the attention weight value will not be overly punished for active users. Although the probabilistic explanation of attention network is broken with $\eta < 1$, we empirically find that this solution leads to performing much better than using the standard softmax.

Because of $\alpha_s$ shouldn’t only consider that each piece of clicked news $r_{si}$ and $d_t$ is neither considered that each piece of clicked ads $r_{tj}$. The aggregated representation $\vec{\alpha}$, $\vec{\beta}$ inspired by the design of MiNet [43] should include

$$
\vec{\alpha}_i = h^T \sigma(W \vec{q}_i \parallel r_{si} \parallel r_{si}^T \parallel M r_{si} \circ q_{ti}), (5)
$$

where $W \in \mathbb{R}^{D_h \times (D_s + D_t + D_u)}$, $h \in \mathbb{R}^{D_h}$, $M \in \mathbb{R}^{D_t \times D_s}$ are parameters to be learned ($D_h$ is a dimension hyper-parameter). $M r_{si} \circ q_{ti}$ denotes the transferred interaction between the clicked news and the target ad. $\circ$ denotes the transfer matrix element-wise product operator. $M$ denotes the delivery information matrix such that $M r_{si}$ can be compared with the target ad $q_{ti}$. $\sigma(\cdot)$ is the rectified linear unit $\text{ReLU}((\text{ReLU}(x) = \max (0, x)))$ as an activation function.

$$
\vec{\beta}_j = h^T \sigma(W \vec{q}_j \parallel r_{idi} \parallel r_{dzi}^T \parallel r_{dzi} \circ q_{tj}), (6)
$$

where $W \in \mathbb{R}^{D_h \times (3D_t + D_u)}$, $h \in \mathbb{R}^{D_h}$ are parameters to be learned. $r_{idi} \circ q_{tj}$ denote the interaction between ad (has clicked) and target ad. They are no need transfer matrix, because it’s located in the same domain.

Next, we need to generate the aggregated feature embedding after obtaining all the user interest vectors. Therefore, we actually form $e_t$ as follows

$$
e_t \triangleq \left[ q_t \parallel v_t p_u \parallel v_s a_s \parallel v_l a_l \right], (7)
$$

$$
e_u \triangleq \left[ q_u \parallel r_{u} \right], (8)
$$

where $v_u$, $v_s$ and $v_t$ are MiNet [11] propose the dynamic weight, $q_s \in \mathbb{R}^{D_s}$ is the cascading of feature embedding dense vectors for the target news. $e_u$ aggregated static user interest from both domains doesn’t use dynamic weight because there is no dimensional deviation.

In particular, dynamic weights as follows

$$
v_u = \text{exp}(g_u^T \text{ReLU}(V_u e_t) + b_u).
$$

$$
v_s = \text{exp}(g_s^T \text{ReLU}(V_s e_t) + b_s).
$$

$$
v_t = \text{exp}(g_t^T \text{ReLU}(V_t e_t) + b_t), (9)
$$

where $V_u \in \mathbb{R}^{D_h \times (D_s + 2D_t + D_u)}$ denotes a matrix parameter, $g_u \in \mathbb{R}^{D_h}$ denotes a vector parameter that projects the hidden state into the attention and $b_u$ denotes a scalar parameter.

3.4 Interaction & Prediction Learning Layer

After we obtain the feature embedding vector $e_t$ & $e_{pu}$ that aggregated user interest, the next task is to exploit non-trivial high-order feature interaction as well as a non-linear knowledge information transformation.
3.4.1 Combination Interaction & Prediction
As mentioned in Section 3.2, plain MLP learns implicit non-trivial high-order feature interactions but its lack of explicit ability to form cross features. And the Cross Network explicitly automatically applies special feature crossing at each layer, efficiently learns high-order combination features by using matrix multiplication, learns highly nonlinear interactions. The idea of Residual Learning [16] is put in place to solve the problem of network performance degradation, which can effectively avoid the problem of gradient disappearance, and can help build a deeper network. Since Cross Net and plain MLP can complement each other in the right way, an intuitive way to make the Cross-Domain model deeper and stronger is tantamount to combine these two components structures. Jointly training Cross Net and MLP components structures together, efficiently learning more information about nonlinear features and combined features, and greatly improve the expressive ability of the model on Cross-Domain recommendation. The resulting model is given in Figure 4. It consists of two implicit feature interactions and explicit feature interactions. The feature embedding vector $e_t$ & $e_{pu}$ that aggregated user interest exploit high-order feature interaction. Finally, we apply the $sigmoid$ function to estimate the click probability of the target ad as

$$y_{t/s} = sigmoid(W^T_{Cross Net}z^l + W^T_{MLP}h^l + b), (10)$$

where $z^l$ are the outputs from the Cross network and $h^l$ are the outputs from the plain MLP, $W$, $b$ are the regression parameters to be learned, $y_{t/s}$ are the output for the target/source domain.

![Figure 4: The architecture of interaction & prediction layer.](image)

3.4.2 Model Cross-Domain Aspect Correlation Learning
Auxiliary Information Enhancement. The complex deep-stacked structure of NAUI may be expected to result in the less-than-perfect outcome. To resolve this result inspired by NON [20] and GoogLeNet [16], we introduce MLP with auxiliary losses to make NAUI easier to train, as shown in Figure 5. The idea behind auxiliary losses is that it makes the intermediate layers more discriminative and improves the discrimination ability of the model. In NAUI MLP part with auxiliary losses, each MLP layer has
an auxiliary loss. Suppose $h^l_i$ denotes the hidden state of $i$-th layer in MLP, and the auxiliary loss of layer $i$ is defined as follows

$$\ell_{aux}^i = \ell\left(\text{sigmoid} \left(W_{aux}^i h^l_i\right), y\right), \quad (11)$$

where $W_{aux}^i$ is model trainable parameters, $\ell(y', y)$ is cross-entropy between $y'$ and $y$, $y$ is the label of a user instance. Then, the loss function of NAUI in the target domain as follows

$$\ell_t = \ell(y', y)_t + \delta \sum_{i} \ell_{aux}^i + \gamma \| W \|_2, \quad (12)$$

where $\delta$ and $\gamma$ are model trainable variables, $y'$ is the final prediction of the top network part, $\| W \|$ is the $L_2$-norm of the network weights. During training, we can adopt different value $\delta$ for auxiliary loss of different layers in MLP and decay $\delta$. We only use the auxiliary loss during training, that is, the network degenerates to plain MLP during validation. In the same way, we have a loss function $\ell_s$ for the same in the source domain.

NAUI model loss. Finally, to train NAUI based on Equation.(2)(in Section 3.2) & Equation.(12) and benefit all domains, with all the model variables are learned in the model by minimizing the combined loss, we defined the optimization objective as follows

$$\ell(\theta) = \ell_t + \epsilon \ell_s, \quad (13)$$

where $\epsilon$ is a balancing hyper-parameter. Model hyper-parameter $\theta$ is $\theta = \theta_t \cup \theta_s \cup \theta_{ru} \cup \theta_{at} \cup \theta_{as}$. As $r_{ru}$ is shared embeddings of bridge users across domains, when optimizing the comprehensive loss, we put $r_{ru}$ learned together based on the data from both domains.

4. Experiments
To validate our intuition, we conduct both offline experiments on two real-world datasets in this section. Comparing the performance of the proposed NAUI as well as the other state-of-the-art recommendation algorithms for CTR prediction.

4.1 Experimental Settings

4.1.1 Datasets
1) UC Toutiao dataset (News-Ads). The UC Toutiao dataset contains four sub-datasets: a random sample of news and ads impression and news clicks history behavior logs and advertisement clicks.
history behavior logs. The target domain is the advertisement and the source domain is the news in the UC dataset. We use history behavior logs for the continuous 6 days (in 2019s) for training, history behavior logs of the next day for validation, and history behavior logs of the day after the next day for testing. After searching for the optimal hyper-parameters on the validation set, we put the initial training set and the validation set as a whole training set.

2) Amazon dataset (Books-Movies). The Amazon product reviews datasets by McAuley et al[32], is a very famous public dataset in real-world to predict the performance of recommender systems. To achieve the cross-domain CTR prediction task, we only utilize the two largest sub-categories: books and movies & TV. where the target domain is the movie & tv and the source domain is the book. We filter the datasets and only keep positive samples(has rating) that have at least five interactions on items with metadata in both domains. We change the data rating scores from 4-5 as label 1 and others scores as label 0. We sort user history logs in chronological order to simulate the actual accuracy of CTR prediction and take out the last rating of each target user for test, the second last rating for validation, and others rating for training. Features include (1) category_ID, (2) items_ID, (3) user reviewed items_ID_list and category_ID_list.

4.1.2 Methods Comparison
We compare the common CTR prediction methods (single-domain and cross-domain). We will compare the following baseline models:

1) Single-Domain CTR Methods
   (1) FM [7]. Factorization Machine (FM) models two situations: 1-order feature importance and 2-order feature interactions for CTR prediction task. FM is regarded as the base model in evaluation.
   (2) DNN[1]. Deep Neural Network (DNN) contains embedding layer & MLP architecture and an output layer. It provides a strong baseline for commonly model comparison.
   (3) Wide & Deep [1]. This method combines LR linear model(wide part) that handles raw features and DNN no-linear model(deep part) that extracts high-order feature interactions.
   (4) DeepFM [2]. A deep model that consists of a “wide” component that models factorization machine saving feature engineering jobs and a deep component.
   (5) DIN [5]. Deep Interest Network model considers the weight of items that models dynamic items interest by designing a local activation unit for CTR prediction task.

2) Cross-Domain Methods
   (6) CoNet [6]. Collaborative cross Network in 2018. It introduces cross-connection units and joint loss functions in MLP to make dual knowledge information transfer possible.
   (7) MiNet [11]. Using an attention mechanism to learn Cross-Domain user interest, and MLP part to learn user high-order feature interaction.
   (8) NAUI. There are the NAUI in this paper.

4.1.3 Parameter Settings
The TensorFlow framework is used to implement all the models. For the fair computing parameter size (because the huge number of distinct features), we assume the dimension size of feature embedding vectors as \(D = 10\). We set \(C = 10\) and the number of MLP layers in models is set to \(L = 3\), the number of cross-layers is set to \(L = 4\), with dimensions are \([1024, 512, 256]\) for UC and \([1024, 512, 256]\) for Amazon. The attention factor of \(D_h\) is set \(D_h = 128\) for UC and \(D_h = 64\) for Amazon. The batch sizes is set to \((512, 128)\) for UC and \((64, 32)\) for Amazon. The optimizer is the Adagrad algorithm. The initial learning rate is 0.001. In addition, we run each method five times and report the average results in our experiments.

4.1.4 Evaluation Metrics
We use two metrics for model evaluation:
UC(Area Under ROC): It reflects the ranking ability of the model. As a commonly used metric in the CTR prediction field. The AUC scores is between [0, 1], where a higher AUC value indicates a better performance. A small improvement may result in a significant increase in CTR prediction recommendation.

Log loss: Since all models attempt to minimize the Log loss over the test set (target domain) defined by Equation.(2). We use cross-entropy as a straightforward metric. The Log loss value is also between [0, 1]. The smaller value indicates better performance.

4.2 Effectiveness of NAUI
Table 1 lists the results on UC dataset and Amazon dataset. It can be seen that in terms of AUC, most of the embedding-based deep neural networks beat the low-order methods such as FM model significantly, implying the importance of modeling high-order feature interactions. FM algorithm performs better than logistic regression algorithm because it models two feature situations: 1-order feature importance and 2-order feature interactions. Among the single-domain prediction approaches, DIN model method performs best. It happens because DIN jointly CIN part that could impact the CTR prediction.

As to the cross-domain CTR prediction approaches, CoNet, MiNet algorithm outperforms shallow modes LR algorithm and factorization machine algorithm. This situation shows that using cross-domain data information can result in improving CTR prediction performance. CoNet introduces cross connection units and joint loss functions in MLP network to enable dual knowledge information transfer across domains. Nevertheless, this method also causes random noise effect and higher computational complexity. MiNet considers user interest for CTR prediction. Our NAUI considers not only the impact of user interest (include static and attentive) but also active user excessive punishment in both domains. By appropriately combining these different signals of interest and adding some auxiliary classifiers, the performance of NAUI is better than other methods.

| Algorithm | UC Toutiao | Amazon |
|-----------|------------|---------|
|           | AUC | Log loss | AUC | Log loss |
| Single-domain | | | | |
| FM | 0.6628 | 0.5247 | 0.7143 | 0.4990 |
| DNN | 0.7156 | 0.4984 | 0.7548 | 0.4593 |
| Wide&Deep | 0.7178 | 0.4979 | 0.7688 | 0.4399 |
| DeepFM | 0.7115 | 0.4998 | 0.7689 | 0.4506 |
| DIN | 0.7141 | 0.4936 | 0.7774 | 0.4490 |
| Cross-domain | | | | |
| CoNet | 0.7165 | 0.4882 | 0.7771 | 0.4489 |
| MiNet | 0.7308 | 0.4351 | 0.7801 | 0.4528 |
| NAUI | **0.7329** | **0.4345** | **0.7861** | **0.4248** |

4.3 Performance Comparison
First, we consider the effect of composite modeling different types of target user interest information learning in NAUI. Figure 6 shows quite diverse phenomena on the two real-world datasets. For the UC Toutiao dataset, modeling user interest (without or smooth) and MLP and aux_loss than modeling user interests and aux_loss with MLP, can lead to much higher AUC. This phenomenon showing that cross features and aux_loss are both quite informative in advertising recommendation. Similarly, for the real-world Amazon dataset, modeling user interest (without or smooth) and Cross Net and aux_loss results in a much higher AUC. This is the nature of rating scores is different from clicks because Amazon is not an advertising dataset, and Amazon is an e-commerce dataset. When these aspects are considered together in NAUI, model can get the highest AUC value. This shows that different types of user interest can be complementary, and joint modeling can produce optimal and more robust performance.
4.4 Hyper-Parameters Study
We will consider the effect of tuning balancing hyper-parameters of NAUI in this section. Figure 7 examine \( \eta \) (smooth hyper-parameters) and Figure 8 examine \( \delta \) (auxiliary loss) respectively. It is observed in the source domain that the results (AUCs) increase when one hyper-parameter enlarges at the beginning but then decreases when its value further enlarges. As we can see, the best smoothing parameter \( \eta \) is 0.5 in the dataset. On the Amazon dataset target domain, large \( \delta \) can lead to very unsatisfactory performance that is results even worse than the target domain only. Overall, the source domain results in a larger AUC improvement than the target domain.

4.5 Effect of the Network Depth
AUC of NAUI with the different number of hidden layers is shown in Figure 9. The hidden layers dimensions \( k \) settings are as follows: 1 layer : \( k = [256] \) dimensions; 2 layers : \( k = [512,256] \); 3 layers : \( k = [1024,256,256] \); 4 layers : \( k = [2048,1024,256,256] \). With the increasing number of hidden layers in NAUI, we can seen that the number of hidden layers can improve the AUC value in the beginning, but when the addition reaches a threshold the advantage is weakened. We also found that adding more hidden layers may lead to performance degradation. This situation possibly due to more model parameters in the Cross-Domains dataset and increased difficulty of training deeper neural networks that lead to optimization difficulties.
5. Conclusion
In this work, we introduced NAUI, which models learning user interest across domains. By adding some parameters to the attention mechanism to reduce the excessive punishment to active users, at the same time, in order to improve the recognition ability of NAUI, the auxiliary classifier and the Cross Net part are added to the interaction of high-order features. Our experiment results have demonstrated the effectiveness of the proposed user interest models and the use of the cross net for learning feature interaction. We believe that NAUI provides another perspective for this interesting and critical task. As future work, we plan to explore NAUI model multi-Cross-Dmains learning.

References
[1] Cheng, H. T. , Koc, L. , Harmsen, J. , Shaked, T. , Chandra, T. , & Aradhye, H. , et al. (2016). Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM.
[2] Guo, H. , Tang, R. , Ye, Y. , Li, Z. , & He, X. . (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. Twenty-Sixth International Joint Conference on Artificial Intelligence.
[3] He, X. , He, Z. , Song, J. , Liu, Z. , Jiang, Y. G. , & Chua, T. S. . (2018). Nais: neural attentive item similarity model for recommendation. IEEE Transactions on Knowledge & Data Engineering, PP(12), 1-1.
[4] Wang, R. , Fu, B. , Fu, G. , & Wang, M. . (2017). Deep & cross network for ad click predictions.
[5] Zhou, G. , Song, C. , Zhu, X. , Fan, Y. , Zhu, H. , & Ma, X. , et al. (2017). Deep interest network for click-through rate prediction.
[6] Hu, G. , Zhang, Y. , & Yang, Q. . (2018). Conet: collaborative cross networks for cross-domain recommendation. Rendle, S. . (2010). Factorization Machines. ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010. IEEE.
[7] Juan, Y. , Lefortier, D. , & Chapelle, O. . (2017). Field-aware Factorization Machines in a Real-world Online Advertising System.
[8] Qu, Y. , Cai, H. , Ren, K. , Zhang, W. , & Wang, J. . (2016). Product-Based Neural Networks for User Response Prediction. 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE.
[9] Lian, J. , Zhang, F. , Xie, X. , & Sun, G. . (2017). CCCFNet: A Content-Boosted Collaborative Filtering Neural Network for Cross Domain Recommender Systems. the 26th International Conference. International World Wide Web Conferences Steering Committee.
[10] Ouyang, W. , Zhang, X. , Zhao, L. , Luo, J. , & Du, Y. . (2020). Minet: mixed interest network for cross-domain click-through rate prediction.
[11] Man, T., Shen, H., Jin, X., & Cheng, X. (2017). Cross-Domain Recommendation: An Embedding and Mapping Approach. *Twenty-Sixth International Joint Conference on Artificial Intelligence*.

[12] Zhou, G., Mou, N., Fan, Y., Pi, Q., & Gai, K. (2019). Deep interest evolution network for click-through rate prediction. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*, 5941-5948.

[13] Elkahky, A. M., Song, Y., & He, X. (2015). A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. *the 24th International Conference. International World Wide Web Conferences Steering Committee*.

[14] Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*. ACM.

[15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., & Rabinovich, A. (2014). Going deeper with convolutions.

[16] Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., & Chua, T. S. (2017). Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*.

[17] Richardson, M., Dominowska, E., & Ragno, R. (2007, May). Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web* (pp. 521-530).

[18] Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018, July). xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1754-1763).

[19] Luo, Y., Zhou, H., Tu, W. W., Chen, Y., Dai, W., & Yang, Q. (2020, July). Network On Network for Tabular Data Classification in Real-world Applications. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2317-2326).

[20] Gao, C., Chen, X., Feng, F., Zhao, K., He, X., Li, Y., & Jin, D. (2019, May). Cross-domain recommendation without sharing user-relevant data. In *The world wide web conference* (pp. 491-502).

[21] Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016, August). Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353-362).

[22] Xiong, C., Wang, T., Ding, W., Shen, Y., & Liu, T. Y. (2012, February). Relational click prediction for sponsored search. In *Proceedings of the fifth ACM international conference on Web search and data mining* (pp. 493-502).

[23] Yin, D., Mei, S., Cao, B., Sun, J. T., & Davison, B. D. (2014, February). Exploiting contextual factors for click modeling in sponsored search. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 113-122).

[24] Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 191-198).

[25] Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z., & Du, Y. (2019, July). Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2078-2086).

[26] Pan, J., Xu, J., Ruiz, A. L., Zhao, W., Pan, S., Sun, Y., & Lu, Q. (2018, April). Field-weighted factorization machines for click-through rate prediction in display advertising. In *Proceedings of the 2018 World Wide Web Conference* (pp. 1349-1357).
[27] He, X., & Chua, T. S. (2017, August). Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 355-364).

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition.

[29] Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. Acoustics Speech & Signal Processing: icassp.international Conference on.

[30] Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., & Schwenk, H., et al. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. Computer Science.

[31] McAuley, J., Targett, C., Shi, Q., & Hengel, A. V. D. (2015). Image-based recommendations on styles and substitutes.