Predicting Different Types of Conversions with Multi-Task Learning in Online Advertising

Junwei Pan
Verizon Media
jwpan@verizonmedia.com

Yizhi Mao
Verizon Media
lolam@verizonmedia.com

Alfonso Lobos Ruiz
UC Berkeley
alobos@berkeley.edu

Yu Sun
Indeed
sunyu@indeed.com

Aaron Flores
Verizon Media
aaron.flores@verizonmedia.com

ABSTRACT

Conversion prediction plays an important role in online advertising since Cost-Per-Action (CPA) has become one of the primary campaign performance objectives in the industry. Unlike click prediction, conversions have different types in nature, and each type may be associated with different decisive factors. In this paper, we formulate conversion prediction as a multi-task learning problem, so that the prediction models for different types of conversions can be learned together. These models share feature representations, but have their specific parameters, providing the benefit of information-sharing across all tasks. We then propose Multi-Task Field-weighted Factorization Machine (MT-FwFM) to solve these tasks jointly. Our experiment results show that, compared with two state-of-the-art models, MT-FwFM improve the AUC by 0.74% and 0.84% on two conversion types, and the weighted AUC across all conversion types is also improved by 0.50%.

CCS CONCEPTS

• **Computing methodologies** → **Factorization methods**; • **Information systems** → **Computational advertising**; • **Theory of computation** → *Computational advertising theory*.

KEYWORDS

Online Advertising, Conversion Prediction, Factorization Machines, Multi-Task Learning

ACM Reference Format:
Junwei Pan, Yizhi Mao, Alfonso Lobos Ruiz, Yu Sun, and Aaron Flores. 2019. Predicting Different Types of Conversions with Multi-Task Learning in Online Advertising. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330783

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330783

1 INTRODUCTION

Online advertising is a 27.5-billion dollar business in fiscal year 2017 [4], and advertisers have been shifting their budgets to programmatic ad buying platforms. Recently, more and more advertisers are running campaigns with Cost-Per-Action (CPA) goals, seeking to maximize conversions for a given budget. To achieve such objectives, accurate prediction of conversion probability is fundamental and has attracted lots of research attention in the past few years [1, 2, 8, 20, 21].

Advertising platforms insert *pixels* (*i.e.* Javascript codes) into advertisers' websites to track users' conversions, and there are several types of conversions that the advertisers want to track. Some pixels track whether a user fills out an online form, while other pixels track whether a user buys a product. The existence of different types of conversions makes conversion prediction challenging because the decisive factors that drive users to convert may vary from one conversion type to another. For example, whether to fill out a form online is a personal decision, so field *User_ID* and its interaction effects with other fields, should be the decisive factor. While for online purchase, the product itself or its corresponding brand play more important roles.

To address this problem, one approach is to build a separate model for each conversion type. However, this is memory intensive and it fails to leverage information from other conversion types. Another approach is to build a unified model which captures the 2-way or 3-way interactions between fields, with conversion type included as one of the fields. However, the 2-way model fails to capture the different field interaction effects for different conversion types, while the 3-way model is computationally expensive.

In this paper we study an alternative approach, *i.e.*, formulating conversion prediction as a multi-task learning problem, so that we can jointly learn prediction models for multiple conversion types. Besides task-specific parameters, these models share low level feature representations, providing the benefit of information sharing among different conversion types. We propose Multi-Task Field-weighted Factorization Machine (MT-FwFM), based on one of the best-performing models for click prediction, *i.e.*, Field-weighted Factorization Machine (FwFM) [24], to solve these tasks together.

Our main contribution is two-fold: First, we formulate conversion prediction as a multi-task learning problem and propose MT-FwFM to solve all tasks jointly. Second, we have carried out extensive experiments on real-world conversion prediction data set to evaluate the performance of MT-FwFM against existing models. The results show that MT-FwFM increases the AUC of ROC on two

conversion types by 0.74% and 0.84%, respectively. The weighted AUC of ROC across all tasks is also increased by 0.50%. We have also conducted comprehensive analysis, which shows that MT-FwFM indeed captures different decisive factors for different conversion types.

The rest of the paper is organized as follows. We investigate the field interaction effects for different conversion types in Section 2. Section 3 describes MT-FwFM in detail. Our experiment results are presented in Section 4. In Section 5, we conduct analysis to show that MT-FwFM learns different field interaction effects for different conversion types. Section 6 and Section 7 discuss the related work and conclude the paper.

2 FIELD INTERACTION EFFECTS FOR DIFFERENT CONVERSION TYPES

The data used for conversion prediction are typically *multi-field categorical data* [32], where features are very sparse and each feature belongs to only one *field*. For example, feature *yahoo.com* and *Nike* belong to field *Page_TLD* (Top-level domain) and *Advertiser*, respectively. In click prediction, it has been verified that different field pairs have different interaction effects on multi-field categorical data [19, 24].

In conversion prediction, advertisers would like to track different types of conversions, and they spend most of their budget on the following four types:

- *Lead*: the user fills out an online form
- *View Content*: the user views a web page such as the landing page or a product page
- *Purchase*: the user purchases a product
- *Sign Up*: the user signs up an account

The decisive factors, *i.e.*, the main effect terms (fields) and/or the interaction terms (field pairs) that drive a user to convert, may vary a lot among these types. Following the analysis in [24], we verify this by computing mutual information (MI) between each field pair and each type of conversion on our real-world data set described later in section 4.1. Suppose there are M unique features $\{x_1, \ldots, x_M\}$, N different fields $\{F_1, \ldots, F_N\}$ and T conversion types. We denote $F(i)$ as the field that feature i belongs to, and $t \in \mathcal{T}$ as the conversion type. The interaction effect of a field pair (F_p, F_q) with respect to conversions of type t is measured by:

$$MI^t((F_p, F_q), Y) = \sum_{(i,j)\in(F_p,F_q)} \sum_{l\in\{0,1\}} p^t((i,j),l) \log \frac{p^t((i,j),l)}{p^t(i,j)p^t(l)} \tag{1}$$

where $p^t((i,j),l)$ is the marginal probability of $p^t(x_i = 1, x_j = 1, y = l)$, $p^t(i,j)$ denotes $p^t(x_i = 1, x_j = 1)$, and $p^t(l)$ is the marginal probability of $p^t(y = l)$. All marginal probabilities are computed based on the samples from each conversion type t.

The top 5 field pairs that have the highest mutual information *w.r.t.* each conversion type are shown in Table 1. It shows that these field pairs vary among types: all 5 field pairs of *Lead* contain field *User_ID* and all 5 field pars of *View Content* contain publisher fields (*Page_TLD* and *Subdomain*)[1]. For *Purchase* and *Sign Up*, most field pairs contain one publisher field and one advertiser field (*Ad*, *Creative*, *Line*). The heat maps of the mutual information for all field pairs with respect to each conversion type are shown in Figure 2 and please refer to section 4.1 for the explanation of each field.

There are several approaches to capture different field interaction effects for different conversion types. The first one is to build one model for each conversion type, and train each model separately. However, this is not preferred in the real-world advertising platform because lots of memories are required to store the parameters of all models. In addition, extreme low conversion rate for some conversion types may render the lack of sufficient positive samples to train the corresponding models.

The second approach is to build a unified model, with conversion type as one of the fields. However, all 2-way state-of-the-art models, such as 2-way Factorization Machines (FM) and Field-weighted Factorization Machines (FwFM), are not able to fully capture the differences in field interaction effects among different conversion types. 3-way FM and FwFM may resolve this issue, but the online computing latency is much higher. Please refer to Section 3.3.3 for the details.

3 MULTI-TASK FIELD-WEIGHTED FACTORIZATION MACHINE

We formulate the prediction of different types of conversions as a multi-task learning problem, and propose Multi-Task Field-weighted Factorization Machine (MT-FwFM) to train these models jointly. This section is organized as follows: Section 3.1 introduces FwFM and MT-FwFM in detail; the training procedure of MT-FwFM is described in section 3.2. In Section 3.3, we analyze the number of parameters as well as computing latency for MT-FwFM.

3.1 Multi-Task Field-weighted Factorization Machine (MT-FwFM)

MT-FwFM is a variant of Field-weighted Factorization Machine (FwFM), which is introduced in [24] for click prediction. FwFM is formulated as

$$p(y|\Theta, \boldsymbol{x}) = \sigma(\Phi_{FwFM}(\Theta, \boldsymbol{x})) \tag{2}$$

where $\sigma(x)$ is the sigmoid function, and $\Phi_{FwFM}(\Theta, \boldsymbol{x})$ is the sum of the main and interaction effects across all features:

$$\begin{aligned}\Phi_{FwFM}(\Theta, \boldsymbol{x}) = w_0 + \sum_{i=1}^{M} x_i \langle \boldsymbol{v}_i, \boldsymbol{w}_{F(i)} \rangle \\ + \sum_{i=1}^{M} \sum_{j=i+1}^{M} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle r_{F(i),F(j)}\end{aligned} \tag{3}$$

Here Θ is a set of parameters $\{w_0, \boldsymbol{w}, \boldsymbol{v}, r\}$: w_0 denotes the bias term; $\boldsymbol{v}_i$ refers to the embedding vector for feature i; $\boldsymbol{w}_{F(i)}$ denotes *main term weight vector* for field $F(i)$, which is used to model the

[1] *Page_TLD* denotes a top-level domain of a web page, while *Subdomain* denotes the subdomain. For example, given a web page with URL *https://sports.yahoo.com/warriors-loss-76ers-vivid-illustration-075301147.html*, the *Page_TLD* is *yahoo.com* and the *Subdomain* is *sports.yahoo.com*

Conversion Type	Top 5 Field Pairs
Lead	(Ad, User), (Creative, User), (Line, User), (Subdomain, User), (Advertiser, User)
View Content	(Subdomain, Hour), (Ad, Subdomain), (Creative, Subdomain), (Subdomain, Age_Bucket), (Page_TLD, Hour)
Purchase	(Ad, Subdomain), (Creative, Subdomain), (Ad, Page_TLD), (Creative, Page_TLD), (Line, Subdomain)
Sign Up	(Ad, Subdomain), (Creative, Subdomain), (Ad, Age_Bucket), (AD, Page_TLD), (Creative, Page_TLD)

Table 1: Top 5 field pairs in terms of mutual information for each conversion type. Please refer to Section 4.1 for the description of these fields in detail.

main effect of feature i; $r_{F(i),F(j)}$ denotes the *field interaction weight* between field $F(i)$ and $F(j)$.

We modify FwFM in the following ways to get MT-FwFM: First, instead of using one bias term w_0, MT-FwFM has one bias term w_0^t for each conversion type t. Second, each conversion type has its own $\boldsymbol{w}_{F(i)}^t$ to model the main effect of feature i. Last, each conversion type also has its own field interaction weights $r_{F(i),F(j)}^t$. The feature embeddings $\boldsymbol{v}_i$ are kept the same as FwFM and are shared by all conversion types. Mathematically,

$$\Phi_{MT-FwFM}(\Theta, \boldsymbol{x}) = w_0^t + \sum_{i=1}^{M} x_i \langle \boldsymbol{v}_i, \boldsymbol{w}_{F(i)}^t \rangle + \sum_{i=1}^{M} \sum_{j=i+1}^{M} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle r_{F(i),F(j)}^t \tag{4}$$

MT-FwFM can be regarded as a 3-layer neural network: each sample is first processed by an *embedding layer* that maps each binary feature x_i to an embedding vector $\boldsymbol{v}_i$, then by a *main & interaction layer* which consists of $\boldsymbol{v}_i$ and $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$.

Each node in the main and interaction layer is connected to a *output layer* which consists of T nodes, one for each conversion type. The connections between $\boldsymbol{v}_i$ and each output node are weighted by $\boldsymbol{w}_{F(i)}^t$, while connections between $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ and each output node are weighted by field interaction weights $r_{F(i),F(j)}^t$. The architecture of MT-FwFM is shown in Figure 1.

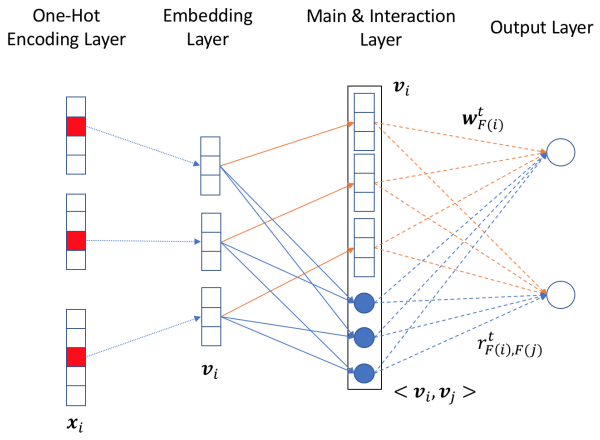


Figure 1: Architecture of MT-FwFM. In the one-hot encoding layer, each field has only one active feature, represented as the red squares. Each active feature is then mapped to an vector in the embedding layer. The blue rectangles in main & interaction layer are copies of the vectors from the embedding layer, while the blue circles represent the dot products between embedding vectors, *i.e.*, $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$. There are T nodes in the output layer, one for each conversion type. The orange dash lines that connect $\boldsymbol{v}_i$ with each output node are weighted by the main term weight vectors $\boldsymbol{w}_{F(i)}^t$, while the connections between blue circles and each output node are weighted by the field interaction weights $r_{F(i),F(j)}^t$. We omitted w_0^t in this figure for the sake of simplicity.

3.2 Joint Training

The feature embedding vectors $\boldsymbol{v}_i$ are shared during the model training by all conversion types and are optimized for every sample. However, for the conversion type specific parameters such as w_0^t, $\boldsymbol{w}_{F(i)}^t$ and $r_{F(i),F(j)}^t$, they are only optimized for samples of corresponding type. We minimize the following loss function for MT-FwFM:

$$\sum_i -y_i \log \hat{y_i} - (1 - y_i) \log(1 - \hat{y_i}) + \lambda \Omega(\Theta) \tag{5}$$

where $\hat{y}_i = p(y|\Theta, \boldsymbol{x}^{(i)})$, y_i denotes the label, and $\Omega(\Theta)$ denotes the regularization terms *w.r.t.* the parameters.

We use mini-batch stochastic gradient descent to optimize the loss function. In each iteration, we select a batch of samples($\mathcal{B}$) randomly, where each sample belongs to a specific task, *i.e.*, conversion type in our case. Within each batch, the model is updated according the conversion type of each sample. More specifically, $\boldsymbol{v}_i$ is updated for all samples, while w_0^t, $\boldsymbol{w}_{F(i)}^t$ and $r_{F(i),F(j)}^t$ are updated only for samples with conversion type t. The training procedure is summarized in Algorithm 1.

Data: $\mathcal{S} = \{(\boldsymbol{x}, y, t)\}$
Initialize parameter $\Theta : \{w, \boldsymbol{v}, \boldsymbol{w}, r\}$ randomly ;
for *epoch* $= 1$ **to** ∞ **do**
  Sample a set of training samples $\mathcal{B}$ from $\mathcal{S}$ randomly
  Compute log loss:
  $L(\Theta) = \sum_{(\boldsymbol{x}, y, t) \in \mathcal{B}} -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) + \lambda \Omega(\Theta)$
  Compute gradient: $\nabla(\Theta)$
  Update model: $\Theta = \Theta - \eta \nabla(\Theta)$
end

Algorithm 1: Training procedure of MT-FwFM.

3.3 Model Complexity

There are two key constraints when we build a conversion prediction model in the real-time serving system: the memory needed to store all parameters, and the computing latency for each sample. We'll analyze these two constraints in this section.

3.3.1 Number of Parameters. The number of parameters in MT-FwFM is

$$T + MK + NTK + \frac{N(N-1)}{2}T \approx MK \tag{6}$$

where T, M, N, K refer to the number of conversion types, features, fields, as well as the dimension of the feature embedding vectors and main term weight vector, respectively.

Thus in (6), T represents the number of bias terms w_0^t; MK calculates the number of parameters for the embedding vectors $\boldsymbol{v}_i$; NTK corresponds to $\boldsymbol{w}_{F(i)}^t$, *i.e.*, the main term weight vectors for all conversion types; $\frac{N(N-1)}{2}T$ denotes the number of field interaction weights $r_{F(i),F(j)}^t$. The number of parameters approximately equals to MK, given that $T \ll M$ and $N \ll M$.

3.3.2 Online Computing Latency. The online computing latency for each prediction request grows linearly with the number of operations, such as float additions and multiplications. During the inference of MT-FwFM, for each sample, the number of operations in the main effect terms is

$$N \cdot (2K - 1) + (N - 1)$$

and the number of operations in the interaction terms is

$$\binom{N}{2} \cdot 2K + \binom{N}{2} - 1$$

Thus the total number of operations of MT-FwFM is

$$N^2K + NK + \binom{N}{2} \approx N^2K$$

3.3.3 MT-FwFM v.s. Using Conversion Type as a Field. Besides formulating conversion prediction as a multi-task learning problem, an alternative approach is to incorporate conversion type as one of the fields in the existing models, such as FM and FwFM. We can either consider the 2-way interactions between fields, referred as *2-way Conversion Type as a Field* (2-way CTF), or the 3-way interactions, referred as *3-way Conversion Type as a Field* (3-way CTF). 2-way CTF with FM and FwFM are used as baseline models in Section 4.

For 3-way CTF with FM or FwFM, the number of operations is much more than that of MT-FwFM, which makes them less preferred in the production environment. We discuss the number of operations of 3-way CTF with FwFM as an example here and omit that for FM since they are very similar. The formula of 3-way CTF with FwFM are:

$$\begin{aligned}\Phi(\Theta, \boldsymbol{x}) = w_0 &+ \sum_{i=1}^{M+T} x_i \langle \boldsymbol{v}_i, \boldsymbol{w}_{F(i)} \rangle \\ &+ \sum_{i=1}^{M+T} \sum_{j=i+1}^{M+T} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \\ &+ \sum_{i=1}^{M} \sum_{j=i+1}^{M} \sum_{t=1}^{T} x_i x_j x_t \langle \boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{v}_t \rangle r_{F(i),F(j)}\end{aligned} \tag{7}$$

where $\langle \boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{v}_t \rangle = \sum_{k=1}^{K} v_i^{(k)} \cdot v_j^{(k)} \cdot v_t^{(k)}$ is a 3-way dot product.

The number of operations of 3-way CTF with FwFM is

$$\left(\frac{5}{2}N^2 + \frac{3}{2}N + 2\right)K + \binom{N}{2} \tag{8}$$

It is approximately $\frac{5}{2}N^2K$, which is 150% more than that of MT-FwFM. Thus, compared with MT-FwFM, 3-way CTF with FwFM is less preferred due to its much more number of operations.

4 EXPERIMENTS

This section presents our experimental evaluation results. We introduce the data set in Section 4.1, and describe the implementation details in Section 4.2. Section 4.3 compares the performance of MT-FwFM with that of 2-way CTF with FM and FwFM. We denote 2-way CTF with FM or FwFM as FM or FwFM in this section for the sake of simplicity.

4.1 Data Set

The data set is collected from the impression and conversion logs of the Verizon Media DSP advertising platform. We treat each impression as a sample, and use the conversions to label them. The labeling is done by *last-touch attribution*, i.e., for each conversion, only the last impression(from the same user and *line* [2]) before this conversion is labeled as a positive sample. All the remaining impressions are labeled as negative samples. The type of each sample is the type of the corresponding line. A line may be associated with multiple conversions that belong to several different types. However, in this paper we focus on those lines that have only one type of conversions since they contribute to most of the traffic as well as spend in our platform.

We use 7 days of impression logs, denoted as T_1 to T_7, as the training data set. Then conversions from T_1 to T_{13} are used to label those impressions. A 6-days longer conversion time window is used because there are usually delays between impressions and conversions, and most conversions happens within 6 days after impressions. We then downsample the negative samples to solve the data imbalance issue since the ratio of positive samples is in the order of 10^{-4} in the data set. We get approximately equal number of positive and negative samples in the training set after downsampling.

The validation data set is collected from the impression logs on T_8, and the test data set is collected on T_9. Conversions from T_8 to T_{14} and T_9 to T_{15} are used to label the validation and test set,

[2]Line is the smallest unit for advertisers to set up budget, goal type, targeting criteria of a group of ads

Data set		Samples	CVR	Features
Train	Purchase	4,552,380	0.1858	11,852
	Lead	6,566,688	0.3402	15,728
	Sign Up	3,332,250	0.8797	13,227
	View Content	170,694	0.3690	1,171
Validation	Purchase	12,800,160	4.63E-04	11,153
	Lead	17,036,604	5.59E-04	9,474
	Sign Up	2,222,334	3.30E-03	5,591
	View Content	441,252	4.90E-04	1,494
Test	Purchase	12,623,382	4.52E-04	11,007
	Lead	18,738,990	5.37E-04	9,373
	Sign Up	1,926,558	3.41E-03	5,553
	View Content	383,940	4.69E-04	1,173

Table 2: Statistics of training, validation and test data sets.

respectively. We do not downsample on validation and test data sets, since the evaluation should be applied to data sets that reflect the real class distribution. Table 2 summarizes the statistics of the training, validation and test data set.

There are 17 fields of features, which fall into 4 categories:

(1) User-side fields: *User_ID*, *Gender* and *Age_Bucket*
(2) Publisher-side fields: *Page_TLD*, *Publisher_ID*, and *Subdomain*
(3) Advertiser-side fields: *Advertiser_ID*, *Creative_ID*, *AD_ID*, *Creative_Media_ID*, *Layout_ID*, and *Line_ID*
(4) Context fields: *Hour_of_Day*, *Day_of_Week*, *Device_Type_ID*, *Ad_Position_ID*, and *Ad_Placement_ID*

We use *Conversion_Type_ID* as an additional field for FM and FwFM. The meanings of most fields are quite straightforward so we only explain some of them:

- *Page_TLD*: top-level domain of a web page.
- *Subdomain*: subdomain of a web page.
- *Creative_ID*: identifier of a creative, which is an image or a video.
- *Ad_ID*: identifier of a (*Line_ID*, *Creative_ID*) combination.
- *Creative_Media_ID*: identifier of the media type of the creative, *i.e.*, image, video or native.
- *Layout_ID*: the size of a creative, for example, 300×200.
- *Device_Type_ID*: identifier of whether this event happens on desktop, mobile or tablet.
- *AD_Position_ID* & *AD_Placement_ID*: identifiers of the position of an ad on the web page.

4.2 Implementations

All baseline models as well as the proposed MT-FwFM model are implemented in Tensorflow. The input is a sparse binary vector $\boldsymbol{x} \in \mathbb{R}^M$ with only N non-zero entries. In the embedding layer, the input vector $\boldsymbol{x}$ is projected into N embedding vectors $\boldsymbol{v}_i$, one for each field. The main and interaction effect terms in the next layer, *i.e.*, main & interaction layer, are computed based on these N vectors. The main effect terms simply concatenate all N vectors, while the interaction effect terms calculate the dot product $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ between each feature pair. Then, each node in the main & interaction layer is connected to the output layer, which consists of T nodes, each of them corresponds to one specific conversion type.

4.3 Performance Comparisons

This section compares MT-FwFM with FM and FwFM on the data sets introduced above. For the hyper-parameters such as regularization coefficient λ and learning rate η in all models, we select the values that lead to the best performance on the validation set and then use them in the evaluation on the test set. We focus on the following performance metrics:

Overall AUC. AUC of ROC (AUC) specifies the probability that, given one positive and one negative sample, their pairwise rank is correct. Overall AUC calculates the AUC over samples from all conversion types.

AUC for each conversion type. The AUC on the samples from each conversion type, denoted as AUC_t.

Weighted AUC. The weighted average of the AUC on each conversion type:

$$\frac{\sum_{t \in \mathcal{T}} AUC_t \cdot N_t}{\sum_{t \in \mathcal{T}} N_t}$$

where N_t refers to the spend of conversion type t. The weights N_t are the spend of each conversion type.

Table 3 summarizes the experiment results. It shows that MT-FwFM gets the best performance *w.r.t.* both overall and weighted AUC, with a lift of 0.19% and 0.50% over the best performing baseline, respectively. While the performance improvement on overall AUC is marginal, the lift on weighted AUC is significant.

Table 4 compares the performance of all models on each conversion type. Among four conversion types, *View Content* and *Purchase* have high AUCs than the other two types using the baseline models(over 95% v.s. under 82%). For these two conversion types that already get high AUC, the lifts of MT-FwFM are more or less neutral, namely 0.03% and −0.24%. On the other hand, for conversion type *Lead* and *Sign Up* that get low performance on baseline models, MT-FwFM improves the AUC by 0.74% and 0.84%.

Therefore, we conclude that MT-FwFM outperforms FM and FwFM significantly *w.r.t.* the weighted AUC over all conversion types. And this improvement mainly come from the conversion types that get relatively low AUC using the baseline models.

5 STUDY OF LEARNED FIELD INTERACTION EFFECTS FOR DIFFERENT CONVERSION TYPES

In this section, we analyze MT-FwFM in terms of its ability to capture different field interaction effects for different conversion types. As described in Section 2, the field interaction effects are measured by the the mutual information between a field pair (F_p, F_q) and the conversion of each type, *i.e.*, $MI^t((F_p, F_q), Y)$. Figure 2 presents the visualization of these field interaction effects by heat maps.

The difference among the four heat maps in Figure 2 illustrates how field interaction effects vary among different conversion types. For *Lead*, *User_ID* has very strong interaction effects with almost all other fields, especially with *Page_TLD*, *Subdomain*, *Ad* and *Creative*. For *View Content*, field pairs containing publisher-side fields such as

Model	Overall AUC			Weighted AUC		
	Training	Validation	Test	Training	Validation	Test
FM	0.9706	0.9014	0.9012	0.9537	0.8500	0.8383
FwFM	0.9702	**0.9023**	0.9027	0.9530	**0.8520**	0.8400
MT-FwFM	**0.9728**	0.8999	**0.9046**	**0.9574**	0.8511	**0.8450**

Table 3: Performance comparison on real-world conversion data set.

Type	Model	Training AUC	Validation AUC	Test AUC
Lead	FM	0.8393	0.8412	0.8116
	FwFM	0.8357	**0.8536**	0.8109
	MT-FwFM	**0.8502**	0.8258	**0.8190**
View Content	FM	0.9523	0.9577	0.9542
	FwFM	0.9511	0.9569	0.9537
	MT-FwFM	**0.9563**	**0.9580**	**0.9545**
Purchase	FM	0.9922	0.9758	0.9684
	FwFM	0.9924	0.9804	**0.9761**
	MT-FwFM	**0.9930**	0.9799	0.9737
Sign Up	FM	0.9381	0.7529	0.7475
	FwFM	0.9374	**0.7564**	0.7501
	MT-FwFM	**0.9428**	0.7545	**0.7585**

Table 4: Performance comparison on data set of each conversion type.

Page_TLD and *Subdomain* have large mutual information in general. For *Purchase* and *Sign Up*, we observe field pairs with advertiser-side fields, such as *Advertiser*, *Ad*, *Creative* and *Line*, have strong interaction effects with other fields.

To verify whether MT-FwFM captures the different patterns of field interaction effects among conversion types, we compare $MI^t((F_p, F_q), Y)$ with the learned field interaction effect between F_p and F_q on conversion type t, namely $|r^t_{F_p,F_q}|$. Here we only consider the magnitude of $r^t_{F_k,F_l}$, since either a large positive or negative value indicate a strong interaction effect. Figure 3 shows the heat maps of $|r^t_{F_p,F_q}|$ for all conversion types.

According to the comparison between Figure 2 and Figure 3, the learned field interaction effects $|r^t_{F_k,F_l}|$ have similar pattern with their mutual information for each conversion type. In general, Figure 3 looks like a pixelated version of Figure 2. For *Lead*, MT-FwFM successfully captures that *User_ID* have strong interaction effects with other fields. For *View Content*, field pairs including the publisher-side fields, *e.g.*, *Publisher*, *Page_TLD*, and *Subdomain* generally have large magnitude of $|r^t_{F_k,F_l}|$. For *Purchase* and *Sign Up*, advertiser-side fields, *e.g.*, *Advertiser*, *Ad*, *Creative* and *Line* have in general large $|r^t_{F_k,F_l}|$ with other fields.

6 RELATED WORK

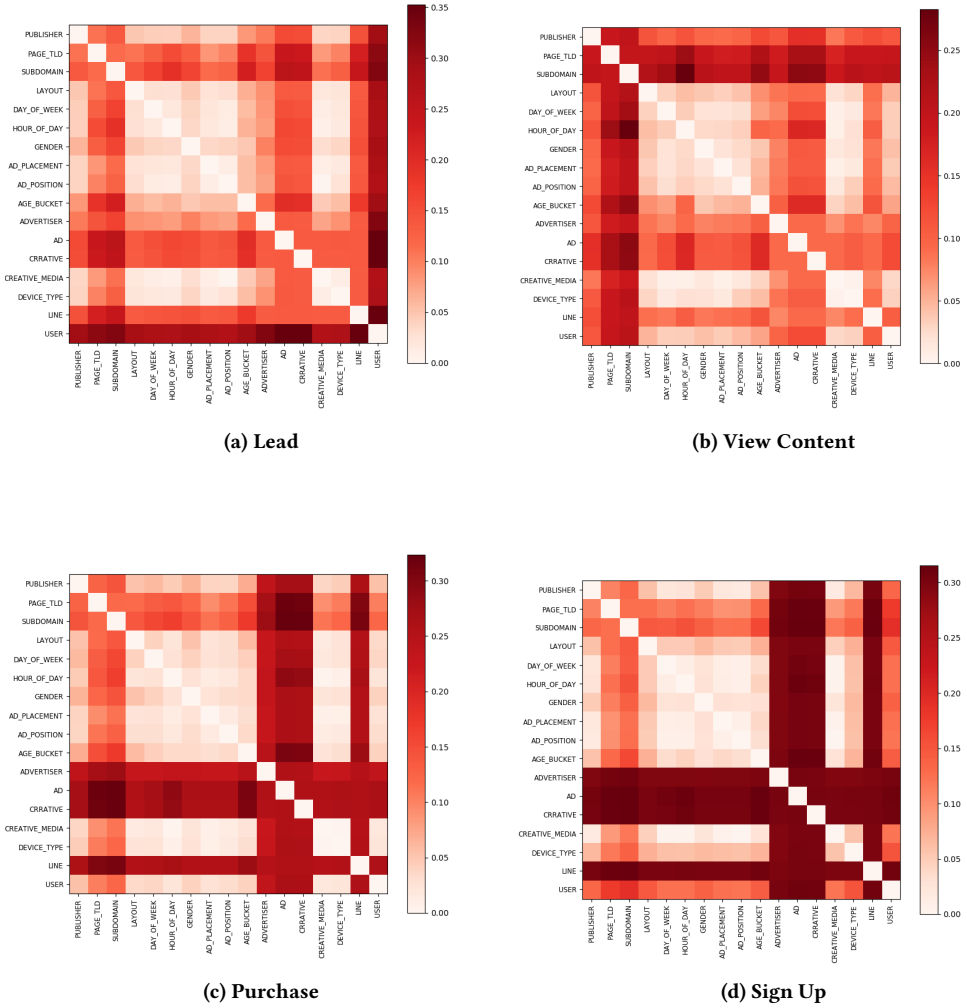
There has been lots of work in the literature on click and conversion prediction in online advertising. Research on click prediction focus on developing various models, including Logistic Regression (LR) [8, 23, 27], Polynomial-2 (Poly2) [6], tree-based models [16], tensor-based models [26], Bayesian models [13], Field-aware Factorization Machines (FFM) [18, 19], and Field-weighted Factorization Machines (FwFM) [24]. Recently, deep learning for CTR prediction also attracted a lot of research attention [9, 14, 15, 25, 29, 30, 32].

For conversion prediction, [20] present an approach to estimate conversion rate based on past performance observations along data hierarchies. [8] and [1] propose a logistic regression model and log-linear model for conversion prediction, respectively. [28] provides comprehensive analysis and proposes a new model for post-click conversion prediction. [3] proposes a ranking model that optimize the conversion funnel even for CPC (Cost-per-Click) campaigns. [17] proposes a time-aware conversion prediction model. [21] describes a practical framework for conversion prediction to tackle several challenges, including extremely sparse conversions, delayed feedback and attribution gaps. Recently, there are also several work on modeling the delay of conversions [7, 31].

Multi-Task Learning (MTL) [5] has been used successfully across multiple applications, from natural language processing [10], speech recognition [11], to computer vision [12]. MTL is also applied to online advertising in [2] to model clicks, conversions and unattributed conversions. In [22] the authors proposes a multi-task model to solve the tasks of click prediction and click-through conversion prediction jointly.

7 CONCLUSION

In this paper, we formulate conversion prediction as a Multi-Task learning problem and propose Multi-Task Field-weighted Factorization Machines (MT-FwFM) to learn prediction models for multiple conversion types jointly. The feature representations are shared by all tasks while each model has its specific parameters, providing the benefit of sharing information among different conversion prediction tasks. Our extensive experiment results show that MT-FwFM



(a) Lead **(b) View Content**

(c) Purchase **(d) Sign Up**

Figure 2: Heat maps of mutual information between field pairs and each type of conversion.

outperforms several state-of-the-art models, including Factorization Machines (FM) and Field-weighted Factorization Machines (FwFM). We also show that MT-FwFM indeed learns different field interaction effects for different conversion types. There are many potential directions for future research. To name a few, we could involve more tasks to the current model, including predicting clicks or non-attributed conversions, or build a deep neural network (DNN) on top of MT-FwFM to better solve these tasks.

REFERENCES

[1] Deepak Agarwal, Rahul Agrawal, Rajiv Khanna, and Nagaraj Kota. 2010. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 213–222.

[2] Amr Ahmed, Abhimanyu Das, and Alexander J Smola. 2014. Scalable hierarchical multitask learning algorithms for conversion optimization in display advertising. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 153–162.

[3] Abraham Bagherjeiran, Andrew O Hatch, and Adwait Ratnaparkhi. 2010. Ranking for the conversion funnel. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 146–153.

[4] Interactive Advertising Bureau. 2017. IAB internet advertising revenue report. https://www.iab.com/wp-content/uploads/2018/05/IAB-2017-Full-Year-Internet-Advertising-Revenue-Report.REV2_.pdf

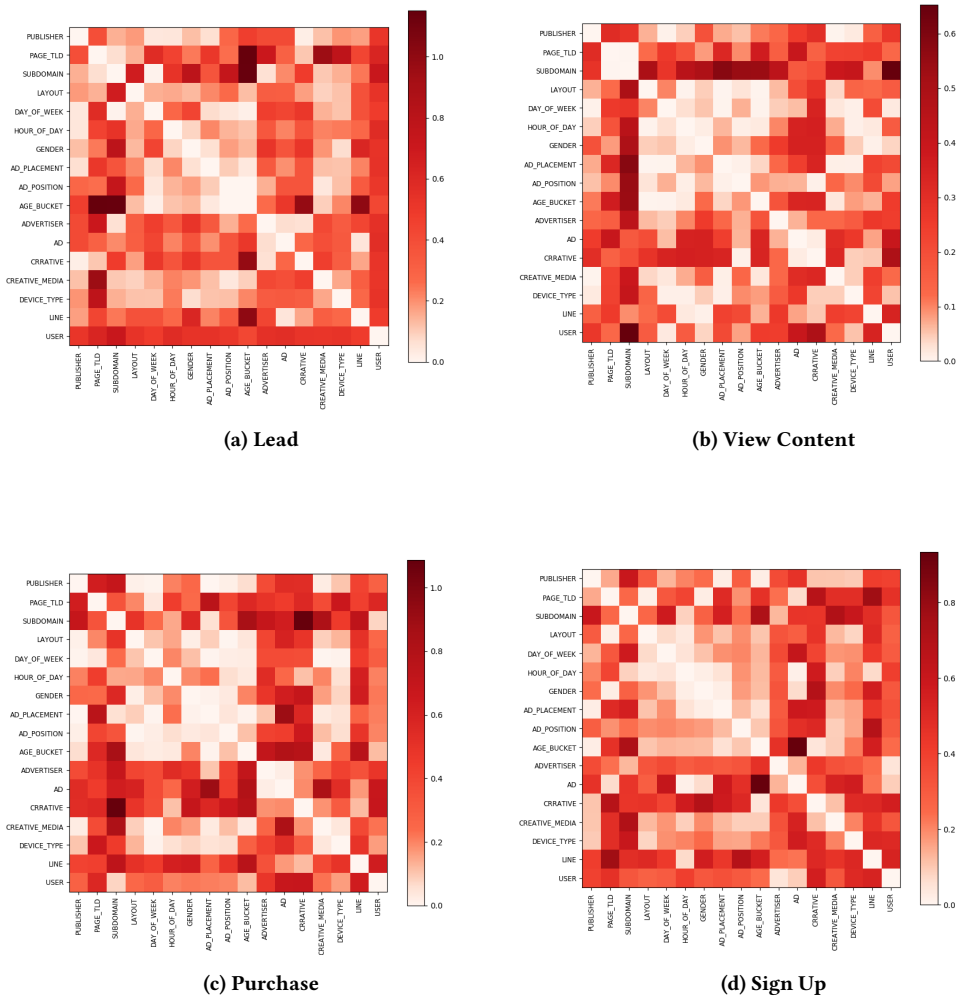
[5] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.

[6] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research* 11, Apr (2010), 1471–1490.

[7] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1097–1105.

[8] Olivier Chapelle, Eren Manavoglu, and Romer Rosales. 2015. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 4 (2015), 61.

[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.



(a) Lead

(b) View Content

(c) Purchase

(d) Sign Up

Figure 3: Heat maps of learned field interaction effects from MT-FwFM, *i.e.*, $|r^t_{F_k,F_l}|$ for different conversion types.

[10] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[11] Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 8599–8603.

[12] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[13] Thore Graepel, Joaquin Q Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 13–20.

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *arXiv preprint arXiv:1703.04247* (2017).

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. (2017).

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

[17] Wendi Ji, Xiaoling Wang, and Feida Zhu. 2017. Time-aware conversion prediction. *Frontiers of Computer Science* 11, 4 (2017), 702–716.

[18] Yuchin Juan, Damien Lefortier, and Olivier Chapelle. 2017. Field-aware factorization machines in a real-world online advertising system. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 680–688.

[19] 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.

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

[21] Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A Practical Framework of Conversion Rate Prediction for Online Display Advertising. In *Proceedings of the ADKDD'17*. ACM, 9.

[22] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-Task Model: An Effective Approach for Estimating

Post-Click Conversion Rate. *arXiv preprint arXiv:1804.07931* (2018).

[23] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1222–1230.

[24] Junwei Pan, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. 2018. Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1349–1357.

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

[26] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 81–90.

[27] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 521–530.

[28] Rómer Rosales, Haibin Cheng, and Eren Manavoglu. 2012. Post-click conversion modeling and analysis for non-guaranteed delivery display advertising. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 293–302.

[29] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC 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.

[30] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. *arXiv preprint arXiv:1708.05123* (2017).

[31] Yuya Yoshikawa and Yusaku Imai. 2018. A Nonparametric Delayed Feedback Model for Conversion Rate Prediction. *arXiv preprint arXiv:1802.00255* (2018).

[32] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*. Springer, 45–57.