Field-weighted Factorization Machines for Click-Through Rate Prediction in Display Advertising

Junwei Pan
Yahoo Research
jwpan@oath.com

Jian Xu
TouchPal Inc.
nayobux@gmail.com

Alfonso Lobos Ruiz
UC Berkeley
alobos@berkeley.edu

Wenliang Zhao
Yahoo Research
wenliangz@oath.com

Shengjun Pan
Yahoo Research
alanpan@oath.com

Yu Sun
LinkedIn Corporation
ysun1@linkedin.com

Quan Lu
Ablibaba Group
quan.lu@gmail.com

ABSTRACT

Click-through rate (CTR) prediction is a critical task in online display advertising. The data involved in CTR prediction are typically multi-field categorical data, i.e., every feature is categorical and belongs to one and only one field. One of the interesting characteristics of such data is that features from one field often interact differently with features from different other fields. Recently, Field-aware Factorization Machines (FFMs) have been among the best performing models for CTR prediction by explicitly modeling such difference. However, the number of parameters in FFMs is in the order of feature number times field number, which is unacceptable in the real-world production systems. In this paper, we propose Field-weighted Factorization Machines (FwFMs) to model the different feature interactions between different fields in a much more memory-efficient way. Our experimental evaluations show that FwFMs can achieve competitive prediction performance with only as few as 4% parameters of FFMs. When using the same number of parameters, FwFMs can bring 0.92% and 0.47% AUC lift over FFMs on two real CTR prediction data sets.

CCS CONCEPTS

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

KEYWORDS

Display Advertising; CTR Prediction; Factorization Machines

ACM Reference Format:
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 *WWW 2018: The 2018 Web Conference, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3178876.3186040

1 INTRODUCTION

Online display advertising is a multi-billion dollar business nowadays, with an annual revenue of 31.7 billion US dollars in fiscal year 2016, up 29% from fiscal year 2015 [2]. One of the core problems display advertising strives to solve is to deliver the right ads to the right people, in the right context, at the right time. Accurately predicting the click-through rate (CTR) is crucial to solve this problem and it has attracted much research attention in the past few years [4, 16, 21].

CLICK	User_ID	GENDER	ADVERTISER	PUBLISHER
1	29127394	Male	Nike	news.yahoo.com
-1	89283132	Female	Walmart	techcrunch.com
-1	91213212	Male	Gucci	nba.com
-1	71620391	Female	Uber	tripadviser.com
1	39102740	Male	Adidas	mlb.com

Table 1: An example of multi-field categorical data for CTR prediction. Each row is an ad impression. Column `CLICK` is the label indicating whether there is a click associated with this impression. Each of the rest columns is a field. Features are all categorical, e.g., `Male`, `Female`, `Nike`, `Walmart`, and each of them belongs to one and only one field, e.g., `Male` belongs to field `GENDER`, and `Nike` belongs to field `ADVERTISER`.

The data involved in CTR prediction are typically *multi-field categorical data* [26] which are also quite ubiquitous in many applications besides display advertising, such as recommender systems [18]. Such data possess the following properties. First, all the features are categorical and are very sparse since many of them are identifiers. Therefore, the total number of features can easily reach millions or tens of millions. Second, every feature belongs to one and only one field and there can be tens to hundreds of fields. Table 1 is an example of a real-world multi-field categorical data set used for CTR prediction.

The properties of multi-field categorical data pose several unique challenges to building effective machine learning models for CTR prediction:

(1) **Feature interactions are prevalent and need to be specifically modeled** [3, 4]. Feature conjunctions usually associate with the labels differently from individual features do. For example, the CTR of `Nike`'s ads shown on `nba.com` is usually much higher than the average CTR of `Nike`'s ads or the average CTR of ads shown on `nba.com`. This phenomenon is usually referred to as *feature interaction* in the literature [12]. To avoid confusion and simplify discussion, in

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186040

the rest of the paper, we unify and abuse the terminology *feature interaction strength* to represent the level of association between feature conjunctions and labels.

(2) **Features from one field often interact differently with features from different other fields**. For instance, we have observed that features from field GENDER usually have strong interactions with features from field ADVERTISER while their interactions with features from field DEVICE_TYPE are relatively weak. This might be attributed to the fact that users with a specific gender are more biased towards the ads they are viewing than towards the type of device they are using.

(3) **Potentially high model complexity needs to be taken care of** [12]. The model parameters such as weights and embedding vectors need to be stored in memory in real-world production systems to enable real-time ad serving. As there are typically millions of features in practice, the model complexity needs to be carefully designed and tuned to fit the model into memory.

To resolve part of these challenges, researchers have built several solutions. Factorization Machines (FMs) [18, 19] and Field-aware Factorization Machines (FFMs) [12, 13] are among the most successful ones. FMs tackle the first challenge by modeling the effect of pairwise feature interactions as the dot product of two embedding vectors. However, the field information is not leveraged in FMs at all. Recently, FFMs have been among the best performing models for CTR prediction and won two competitions hosted by Criteo and Avazu [13]. FFMs learn different embedding vectors for each feature when the feature interacts with features from different other fields. In this way, the second challenge is explicitly tackled. However, the number of parameters in FFMs is in the order of feature number times field number, which can easily reach tens of millions or even more. This is unacceptable in real-world production systems. In this paper, we introduce Field-weighted Factorization Machines (FwFMs) to resolve all these challenges simultaneously. The main contributions of this paper can be summarized as follows:

(1) Empirically we show that the average interaction strength of feature pairs from one field pair tends to be quite different from that of other field pairs. In other words, different field pairs have significantly different levels of association with the labels (i.e. clicks in CTR prediction). Following the same convention, we call this *field pair interactions*.

(2) Based on the above observation, we propose Field-weighted Factorization Machines (FwFMs). By introducing and learning a field pair weight matrix, FwFMs can effectively capture the heterogeneity of field pair interactions. Moreover, parameters in FwFMs are magnitudes fewer than those in FFMs, which makes FwFMs a preferable choice in real-world production systems.

(3) FwFMs are further augmented by replacing the binary representations of the linear terms with embedding vector representations. This novel treatment can effectively help avoid over-fittings and enhance prediction performance.

(4) We conduct comprehensive experiments on two real-world CTR prediction data sets to evaluate the performance of FwFMs against existing models. The results show that FwFMs can achieve competitive prediction performance with only as few as 4% parameters of FFMs. When using the same number of parameters, FwFMs outperform FFMs by up to 0.9% AUC lift.

The rest of the paper is organized as follows. Section 2 provides the preliminaries of existing CTR prediction models that handle multi-field categorical data. In Section 3, we show that the interaction strengths of different field pairs are quite different, followed by detailed discussion of the proposed model in Section 4. Our experimental evaluation results are presented in Section 5. In Section 6, we show that FwFMs learn the field pair interaction strengths even better than FFMs. Section 7 and Section 8 discusses the related work and concludes the paper respectively.

2 PRELIMINARIES

Logistic Regression (LR) is probably the most widely used machine learning model on multi-field categorical data for CTR prediction [4, 21]. Suppose there are m unique features $\{f_1, \cdots, f_m\}$ and n different fields $\{F_1, \cdots, F_n\}$. Since each feature belongs to one and only one field, to simplify notation, we use index i to represent feature f_i and $F(i)$ to represent the field f_i belongs to. Given a data set $S = \{y^{(s)}, \boldsymbol{x}^{(s)}\}$, where $y^{(s)} \in \{1, -1\}$ is the label indicating click or non-click and $\boldsymbol{x}^{(s)} \in \{0, 1\}^m$ is the feature vector in which $x_i^{(s)} = 1$ if feature i is active for this instance otherwise $x_i^{(s)} = 0$, the LR model parameters $\boldsymbol{w}$ are estimated by minimizing the following regularized loss function:

$$\min_{\boldsymbol{w}} \ \lambda \|\boldsymbol{w}\|_2^2 + \sum_{s=1}^{|S|} \log(1 + \exp(-y^{(s)} \Phi_{LR}(\boldsymbol{w}, \boldsymbol{x}^{(s)}))) \tag{1}$$

where λ is the regularization parameter, and

$$\Phi_{LR}(\boldsymbol{w}, \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i \tag{2}$$

is a linear combination of individual features.

However, linear models are not sufficient for tasks such as CTR prediction in which feature interactions are crucial [4]. A general way to address this problem is to add feature conjunctions. It has been shown that Degree-2 Polynomial (Poly2) models can effectively capture the effect of feature interactions[3]. Mathematically, in the loss function of equation (1), Poly2 models consider replacing Φ_{LR} with

$$\Phi_{Poly2}(\boldsymbol{w}, \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j w_{h(i,j)} \tag{3}$$

where $h(i, j)$ is a function hashing i and j into a natural number in hashing space H to reduce the number of parameters. Otherwise the number of parameters in the model would be in the order of $O(m^2)$.

Factorization Machines(FMs) learn an embedding vector $\boldsymbol{v}_i \in \mathbb{R}^K$ for each feature, where K is a hyper-parameter and is usually a small integer, e.g., 10. FMs model the interaction between two features i and j as the dot product of their corresponding embedding vectors $\boldsymbol{v}_i, \boldsymbol{v}_j$:

$$\Phi_{FMs}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \quad (4)$$

FMs usually outperform Poly2 models in applications involving sparse data such as CTR prediction. The reason is that FMs can always learn some meaningful embedding vector for each feature as long as the feature itself appears enough times in the data, which makes the dot product a good estimate of the interaction effect of two features even if they have never or seldom occurred together in the data. However, FMs neglect the fact that a feature might behave differently when it interacts with features from different other fields. Field-aware Factorization Machines (FFMs) model such difference explicitly by learning $n-1$ embedding vectors for each feature, say i, and only using the corresponding one $\boldsymbol{v}_{i,F(j)}$ to interact with another feature j from field $F(j)$:

$$\Phi_{FFMs}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j \langle \boldsymbol{v}_{i,F(j)}, \boldsymbol{v}_{j,F(i)} \rangle \quad (5)$$

Although FFMs have got significant performance improvement over FMs, their number of parameters is in the order of $O(mnK)$. The huge number of parameters of FFMs is undesirable in the real-world production systems. Therefore, it is very appealing to design alternative approaches that are competitive and more memory-efficient.

3 INTERACTION STRENGTHS OF FIELD PAIRS

In multi-field categorical data, every feature belongs to one and only one field. We are particularly interested in whether the strength of interactions are different at the field level. In other words, whether the average interaction strength between all feature pairs from a field pair is different from that of other field pairs.

For example, in the CTR prediction data, features from field `ADVERTISER` usually have strong interaction with features from field `PUBLISHER` since advertisers usually target a group of people with specific interest and the audience of publishers are naturally grouped by interests. On the other hand, features from field `HOUR_OF_DAY` tend to have little interaction with features from field `DAY_OF_WEEK`, which is not hard to understand since by intuition their interactions reveal little about the clicks.

To validate the heterogeneity of the field pair interactions, we use mutual information [6] between a field pair (F_k, F_l) and label variable Y to quantify the interaction strength of the field pair:

$$MI((F_k, F_l), Y) = \sum_{(i,j) \in (F_k, F_l)} \sum_{y \in Y} p((i,j), y) log \frac{p((i,j), y)}{p(i,j)p(y)} \quad (6)$$

Figure 1 is a visualization of the mutual information between each field pair and the label, computed from Oath CTR data. Unsurprisingly, the interaction strengths of different field pairs are quite different. Some field pairs have very strong interactions, such as (`AD_ID`, `SUBDOMAIN`), (`CREATIVE_ID`, `PAGE_TLD`) while some other field pairs have very weak interactions, such as (`LAYOUT_ID`, `GENDER`), (`DAY_OF_WEEK`, `AD_POSITION_ID`). The meanings of the these fields are explained in Section 5.1

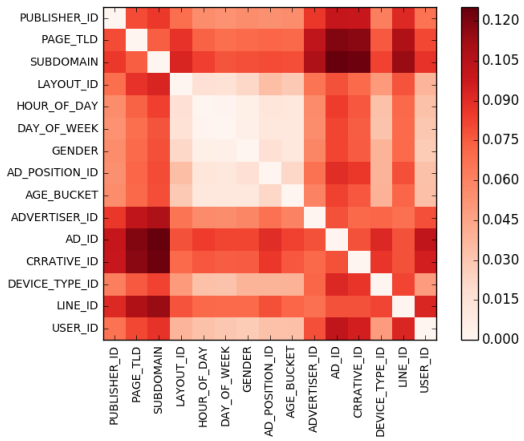


Figure 1: Heat map of mutual information between each field pair and the label.

Although the analysis result is not surprising, we note that none of the existing models takes this field level interaction heterogeneity into consideration. This motivated us to build an effective machine learning model to capture the different interaction strengths of different field pairs.

4 FIELD-WEIGHTED FACTORIZATION MACHINES (FWFMS)

We propose to explicitly model the different interaction strengths of different field pairs. More specifically, the interaction of a feature pair i and j in our proposed approach is modeled as

$$x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle r_{F(i),F(j)}$$

where $\boldsymbol{v}_i, \boldsymbol{v}_j$ are the embedding vectors of i and j, $F(i), F(j)$ are the fields of feature i and j, respectively, and $r_{F(i),F(j)} \in \mathbb{R}$ is a weight to model the interaction strength between field $F(i)$ and $F(j)$. We refer to the resulting model as the Field-weighted Factorization Machines(FwFMs):

$$\Phi_{FwFMs}((\boldsymbol{w}, \boldsymbol{v}), \boldsymbol{x}) = w_0 + \sum_{i=1}^{m} x_i w_i + \sum_{i=1}^{m} \sum_{j=i+1}^{m} x_i x_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle r_{F(i),F(j)} \quad (7)$$

FwFMs are extensions of FMs in the sense that we use additional weight $r_{F(i),F(j)}$ to explicitly capture different interaction strengths of different field pairs. FFMs can model this implicitly since they learn several embedding vectors for each feature i, each one $\boldsymbol{v}_{i,F_k}$ corresponds to one of other fields $F_k \neq F(i)$, to model its different interaction with features from different fields. However, the model complexity of FFMs is significantly higher than that of FMs and FwFMs.

4.1 Model Complexity

The number of parameters in FMs is $m + mK$, where m accounts for the weights for each feature in the linear part $\{w_i | i = 1, ..., m\}$ and mK accounts for the embedding vectors for all the features

Model	Number of Parameters
LR	m
Poly2	$m + H$
FMs	$m + mK$
FFMs	$m + m(n-1)K$
FwFMs	$m + mK + \frac{n(n-1)}{2}$

Table 2: A summary of model complexities (ignoring the bias term w_0). m and n are feature number and field number respectively, K is the embedding vector dimension, and H is the hashing space size when hashing tricks are used for Poly2 models.

Data set		Samples	Fields	Features
Criteo	Train	27,502,713	26	399,784
	Validation	9,168,820	26	399,654
	Test	9,169,084	26	399,688
Oath	Train	24,885,731	15	156,401
	Validation	7,990,874	15	101,217
	Test	8,635,361	15	100,515

Table 3: Statistics of training, validation and test sets of Criteo and Oath data sets respectively.

$\{\boldsymbol{v}_i | i = 1, ..., m\}$. FwFMs use $n(n-1)/2$ additional parameters $\{r_{F_k,F_l} | k, l = 1, ..., n\}$ for each field pair so that the total number of parameters of FwFMs is $m + mK + n(n-1)/2$. For FFMs, the number of parameters is $m + m(n-1)K$ since each feature has $n-1$ embedding vectors. Given that usually $n \ll m$, the parameter number of FwFMs is comparable with that of FMs and significantly less than that of FFMs. In Table 2 we compare the model complexity of all models mentioned so far.

4.2 Linear Terms

In the linear terms $\sum_{i=1}^{m} x_i w_i$ of equation (7), we learn a weight w_i for each feature to model its effect with the label, using binary variable x_i to represent feature i. However, the embedding vectors $\boldsymbol{v}_i$ learned in the interaction terms should capture more information about feature i, therefore we propose to use $x_i \boldsymbol{v}_i$ to represent each feature in the linear terms as well.

We can learn one linear weight vector $\boldsymbol{w}_i$ for each feature and the linear terms become:

$$\sum_{i=1}^{m} x_i \langle \boldsymbol{v}_i, \boldsymbol{w}_i \rangle \tag{8}$$

There are mK parameters in the feature-wise linear weight vectors, and the total number of parameters is $2mK + \frac{n(n-1)}{2}$. Alternatively, we can learn one linear weight vector $\boldsymbol{w}_{F(i)}$ for each field and all features from the same field $F(i)$ use the same linear weight vector. Then the linear terms can be formulated as:

$$\sum_{i=1}^{m} x_i \langle \boldsymbol{v}_i, \boldsymbol{w}_{F(i)} \rangle \tag{9}$$

The parameter number of these kind of FwFMs is $nK + mK + \frac{n(n-1)}{2}$, which is almost the same as FwFMs with original linear weights since both K and n are usually in the order of tens.

In the rest of the paper, we denote FwFMs with original linear weights as FwFMs_LW, denote FwFMs with feature-wise linear weight vectors as FwFMs_FeLV, denote FwFMs with field-wise linear weight vectors as FwFMs_FiLV.

5 EXPERIMENTS

In this section we present our experimental evaluations results. We will first describe the data sets and implementation details in Section 5.1 and 5.2 respectively. In Section 5.3.1 we compare FwFMs_LW, i.e., FwFMs using original linear weights, with LR, Poly2, FMs and FFMs. Then we further investigate the performance of FwFMs_LW and FFMs when they use the same number of parameters in Section 5.3.2. The enhancement brought by our novel linear term treatment is discussed in Section 5.3.3. Finally we show model hyper-parameter tuning details in Section 5.4.

5.1 Data sets

We use the following two data sets in our experiments.

(1) Criteo CTR data set: This is the data set used for the Criteo Display Advertising Challenge [15]. We split the data into training, validation and test sets randomly by 60%:20%:20%.
(2) Oath CTR data set: We use two-week display advertising click log from our ad serving system as the training set and the log of next day and the day after next day as validation and test set respectively.

The Criteo data set is already label balanced. For the Oath CTR data set, the ratio of positive samples (clicks) is the overall CTR, which is typically smaller than 1%. We downsample the negative examples so that the positive and negative samples are more balanced. The downsampling is not done for validation and test sets, since the evaluation should be applied to data sets reflecting the actual traffic distribution.

There are 26 anonymous categorical fields in Criteo data set. The Oath data set consists of 15 fields, which can be categorized into 4 groups: user side fields such as `GENDER`, `AGE_BUCKET`, `USER_ID`, publisher side fields such as `PAGE_TLD`, `PUBLISHER_ID`, `SUBDOMAIN`, advertiser side fields such as `ADVERTISER_ID`, `AD_ID`, `CREATIVE_ID`, `LAYOUT_ID` , `LINE_ID` and context side fields such as `HOUR_OF_DAY`, `DAY_OF_WEEK`, `AD_POSITION_ID` and `DEVICE_TYPE_ID`. The meanings of most fields are quite straightforward and we only explain some of them: `PAGE_TLD` denotes the top level domain of a web page and `SUBDOMAIN` denotes the sub domain of a web page. `CREATIVE_ID` denotes the identifier of a creative, while `AD_ID` identifies an ad that encapsulates a creative; the same creative may be assigned to different ads. `DEVICE_TYPE_ID` denotes whether this events happens on desktop, mobile or tablet. `LAYOUT_ID` denotes a specific ads size and `AD_POSITION_ID` denotes the position for ads in web page.

Furthermore, for both data sets we filter out all features which appear less than τ times in the training set and replace them by a `NULL` feature, where τ is set to 20 for Criteo data set and 10 for Oath data set. The statistics of the two data sets are shown in Table 3.

5.2 Implementations

We use LibLinear [7] to train Poly2 with hashing tricks [25] to hash feature conjunctions to a hashing space of 10^7. All the other models are implemented in Tensorflow. We follow the implementation of LR and FMs in [17], and implement FFMs by ourselves. The architecture of FwFMs in Tensorflow is shown in Figure 2.

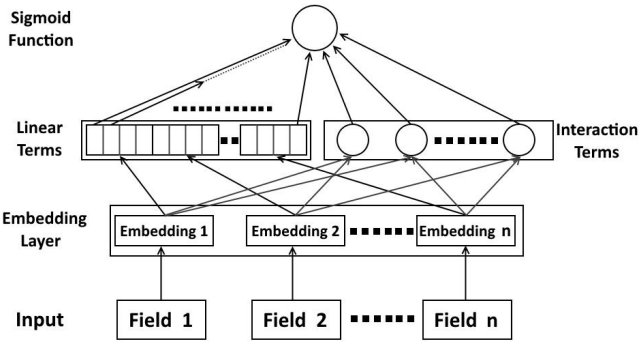


Figure 2: Implementation for FwFMs in Tensorflow.

The input is a sparse binary vector $\boldsymbol{x}_i \in \mathbb{R}^m$ with only n non-zero entries since there are n fields and each field has one and only one active feature for each sample. In the embedding layer, the input vector $\boldsymbol{x}_i$ is projected into n embedding vectors. In the next layer the linear terms and interaction terms are computed from these n latent vectors. The linear terms layer simply concatenates the latent vectors of all active features. The interaction terms layer calculates the dot product $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ between embedding vectors of all pairs of active features. Then every node in the linear terms layer and interaction terms layer will be connected to the final output node, which will sum up the inputs from both linear terms layer and interaction terms layer with weights. Note that in FwFMs_FeLV, the weights between the linear terms layer and the final output is w_i while for FwFMs_FeLV and FwFM_FiLV the weights are $\boldsymbol{w}_i$ and $\boldsymbol{w}_{F(i)}$ respectively. The weights between the interaction terms layer and the output node are $r_{F(i),F(j)}$.

5.3 Performance Comparisons

5.3.1 Comparison of FwFMs with Existing Models. In this section we will conduct performance evaluation for FwFMs with original linear terms, i.e., FwFMs_LW. We will compare it with LR, Poly2, FMs and FFMs on two data sets mentioned above. For the parameters such as regularization coefficient λ, and learning rate η in all models, we select those which leads to the best performance on the validation set and then use them in the evaluation on test set. Experiment results can be found in Table 4.

We observe that FwFMs can achieve better performance than LR, Poly2 and FMs on both data sets. The improvement comes from the fact that FwFMs explicitly model the different interaction strengths of field pairs, which would be discussed in Section 6 in more details. FFMs can always get the best performance on training and validation sets. However, FwFMs' performance on the both test sets are quite competitive with that of FwFMs. It suggests that FFMs are more vulnerable to overfittings than FwFMs.

5.3.2 Comparison of FwFMs and FFMs using the same number of parameters. One critical drawback of using FFMs is that their number of parameters is in the order of $O(mnK)$, which would be too large to fit in memory. There are two solutions to reduce the number of parameters in FFMs [12]: use a smaller K or use hashing tricks with a small hashing space H. Juan et. al. [12] proposed to use $K_{FFMs} = 2$ for FFMs to get a good trade-off between prediction accuracy and the number of parameters. This reduce the number of parameters to $(n-1)mK_{FFMs}$. Juan et. al. [12] also proposed to further reduce the number of parameters of FFMs by using hashing tricks with a small hashing space H_{FFMs} to reduce it to $(n-1)m_{FFMs}K_{FFMs}$. In this section, we make a fair comparison of FwFMs with FFMs using the same number of parameters by choosing proper k_{FFMs} and H_{FFMs} so that the number of parameters of FFMs and FwFMs are the same, i.e., $(n-1)H_{FFMs}K_{FFMs} = mK_{FwFMs}$, we do not count the parameters from the linear terms since they are negligible compared with the number of interaction terms. We choose $K_{FwFMs} = 10$ and $K_{FFMs} = 2$ and $K_{FFMs} = 4$ as described in [12]. The experimental results are shown in Table 5.

We observe that when using the same number of parameters, FwFMs get better performance on test sets of both Criteo and Oath data sets, with a lift of 0.70% and 0.45% respectively. We conclude that FFMs make lots of compromise on the prediction performance when reducing the feature numbers therefore FwFMs can outperform them significantly in such cases.

5.3.3 FwFMs with Different Linear Terms. We conduct experiments to compare FwFMs with three different kinds of linear terms as mentioned in Section 4.2. Table 6 lists the performance comparison between them. We observe that, FwFMs_LW and FwFMs_FeLV can achieve better performance on the training and validation set than FwFMs_FiLV. The reason is that those two models have more number of parameters in the linear weights(m and mK) than FwFMs_FiLV(nK), so they can fit the training and validation set better than FwFMs_FiLV. However, FwFMs_FiLV get the best results on the test set, which suggests that it has better generalization performance. Furthermore, when we compare FwFMs_FiLV with FFMs using the same number of parameters, the AUC lift on Oath data set and Criteo data set are 0.92% and 0.47% respectively.

5.4 Hyper-parameter Tuning

In this section we will show the impact of regularization coefficient λ, embedding vector dimension K and learning rate η on FwFMs. All following evaluations are done for FwFMs_FiLV model on Oath validation set.

5.4.1 Regularization. We add L_2 regularizations of all parameters in FwFMs to the loss function to prevent over-fitting. Figure 3 shows the AUC on validation set using different λ. We get the best performance on validation set using $\lambda = 1e-5$.

5.4.2 Learning Rate and Embedding Vector Dimension. We have done experiments to check the impact of learning rate η to the performance of FwFMs, and the results are shown in Figure 4. It shows that by using a small η, we can keep improving the performance on validation set slowly in the first 20 epochs, while using a large η will improve the performance quickly and then lead to over-fitting. In all experiments on Oath data set we choose $\eta = 1e-4$.

Models	Parameters	AUC		
		Training	Validation	Test
LR	$\eta = 1e-4, \lambda = 1e-7, t = 15$	0.8595	0.8503	0.8503
Poly2	$s = 7, c = 2$	0.8652	0.8542	0.8523
FMs	$\eta = 5e-4, \lambda = 1e-6, k = 10, t = 10$	0.8768	0.8628	0.8583
FFMs	$\eta = 1e-4, \lambda = 1e-7, k = 10, t = 3$	**0.8833**	**0.8660**	**0.8624**
FwFMs	$\eta = 1e-4, \lambda = 1e-5, k = 10, t = 15$	0.8827	0.8659	0.8614

(a) Oath data set

Models	Parameters	AUC		
		Training	Validation	Test
LR	$\eta = 5e-5, \lambda = 1e-6, t = 14$	0.7716	0.7657	0.7654
Poly2	$s = 7, c = 2$	0.7847	0.7718	0.7710
FMs	$\eta = 1e-4, \lambda = 1e-6, k = 10, t = 10$	0.7925	0.7759	0.7761
FFMs	$\eta = 5e-4, \lambda = 1e-7, k = 10, t = 3$	**0.7989**	**0.7781**	**0.7768**
FwFMs	$\eta = 1e-4, \lambda = 1e-6, k = 10, t = 8$	0.7941	0.7772	0.7764

(b) Criteo data set

Table 4: Comparison among models on Criteo and Oath CTR data sets.

Model	Oath data set			Criteo data set		
	Training	Validation	Test	Training	Validation	Test
FFMs($K = 2, H = \frac{10}{14 \cdot 2}m$)	0.8743	0.8589	0.8543	0.7817	0.7716	0.7719
FFMs($K = 4, H = \frac{10}{14 \cdot 4}m$)	0.8708	0.8528	0.8418	0.7697	0.7643	0.7641
FwFMs	**0.8827**	**0.8659**	**0.8614**	**0.7941**	**0.7772**	**0.7764**

Table 5: Comparison of FFMs with FwFMs using same number of parameters. K is embedding vector dimension and H is the hashing space for hashing tricks.

Model	Oath data set			Criteo data set		
	Training	Validation	Test	Training	Validation	Test
FwFMs_LW	**0.8827**	0.8659	0.8614	0.7941	0.7772	0.7764
FwFMs_FeLV	0.8829	**0.8665**	0.8623	**0.7945**	**0.7774**	0.7763
FwFMs_FiLV	0.8799	0.8643	**0.8635**	0.7917	0.7766	**0.7766**

Table 6: Performance of FwFMs with different linear terms.

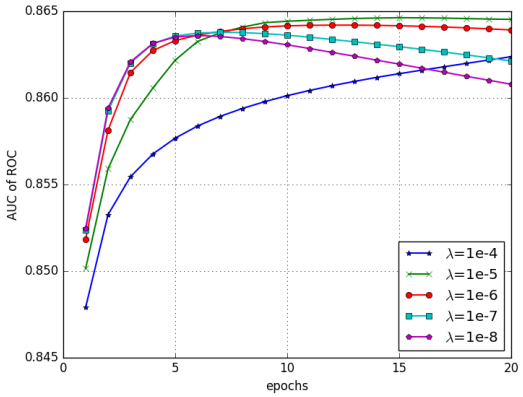


Figure 3: Impact of regularization coefficient λ on FwFMs.

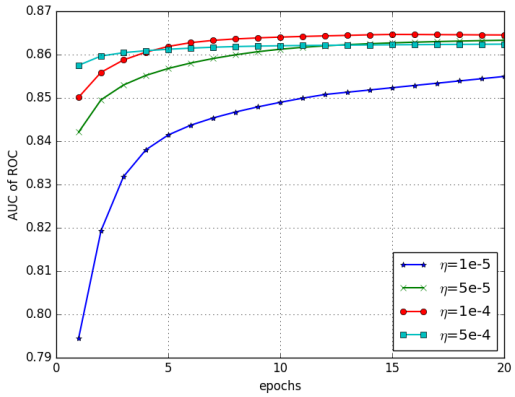


Figure 4: Impact of learning rate η on FwFMs.

k	Train AUC	Validation AUC	Training time (s)
5	0.8794	0.8621	320
10	0.8827	0.8659	497
15	0.8820	0.8644	544
20	0.8822	0.8640	636
30	0.8818	0.8647	848
50	0.8830	0.8652	1113
100	0.8830	0.8646	1728
200	0.8825	0.8646	3250

Table 7: Comparison of different embedding vector dimensions K

We conduct experiments to investigate the impact of embedding vector dimension K on the performance of FwFMs. Table 7 shows that the AUC changes only a little when we use different K. We choose $K = 10$ in FwFMs since it gets best trade-off between performance and training time.

6 STUDY OF LEARNED FIELD INTERACTION STRENGTHS

In this section, we compare FMs, FFMs and FwFMs in terms of their ability to capture interaction strengths of different field pairs. Our result shows that FwFMs can model the interaction strength much better than FMs and FFMs do thanks to the field pair interaction weight r_{F_k,F_l}.

In Section 3 the interaction strengths of field pairs are quantified by mutual information and visualized in heat map (Figure 5a). To measure the learned field interaction strengths, we define the following metric:

$$\frac{\sum_{(i,j)\in(F_k,F_l)} I(i,j)\cdot \#(i,j)}{\sum_{(i,j)\in(F_k,F_l)} \#(i,j)} \tag{10}$$

where $\#(i,j)$ is the number of times feature pair (i,j) appears in the training data, $I(i,j)$ is the learned strength of interaction between feature i and j. For FMs $I(i,j) = |\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle|$, for FFMs $I(i,j) = |\langle \boldsymbol{v}_{i,F_l}, \boldsymbol{v}_{j,F_k} \rangle|$, for FwFMs $I(i,j) = |\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \cdot r_{F_k,F_l}|$. Note that we sum up the absolute values of the inner product terms otherwise positive values and negative values would counteract with each others.

If a model can capture the interaction strengths of different field pairs well, we expect its learned field interaction strengths to be close to the mutual information as shown in equation (6). To facilitate the comparison, in Figure 5 we plot the field interaction strengths learned by FMs, FFMs and FwFMs in the form of heatmap, along with heatmap of the mutual information between field pairs and labels. We also computed Pearson correlation coefficient to quantitatively measure how well the learned field interaction strengths match the mutual information.

From Figure 5b we observe that FMs learned interaction strengths quite different from mutual information. Some field pairs which have high mutual information with labels only get low interaction strengths, for example (AD_ID, SUBDOMAIN), (CREATIVE_ID, PAGE_TLD). While some field pairs with low mutual information have high interaction strengths, such as (LAYOUT_ID, GENDER). The Pearson correlation coefficient between the learned field pair interaction strengths and mutual information is -0.1210. This is not surprising since FMs is not considering field information when modeling feature interaction strengths.

As shown in Figure 5c, FFMs are able to learn interaction strengths more similar to mutual information. This is confirmed by a higher Pearson correlation coefficient of 0.1544, which is much better than FMs.

For FwFMs, Figure 5d shows that the heat map of the learned interaction strengths are very similar to that of mutual information. For example, the field pairs that have highest mutual informations with labels, such as (AD_ID,SUBDOMAIN), (AD_ID,PAGE_TLD), and (CREATIVE_ID,PAGE_TLD), also have higher interaction strengths. For fields such as LAYOUT_ID, HOUR_OF_DAY, DAY_OF_WEEK, GENDER and AD_POSITION_ID, field pairs between any of them have low mutual information with labels as well as low interaction strengths. The Pearson correlation coefficient with mutual information is 0.4271, which is much better than FMs and FFMs.

To understand the importance of the weights r_{F_k,F_l} in modeling the interaction strength of field pairs, we plot the heap map of r_{F_k,F_l}, as well as learned field interaction strengths of FwFMs without r_{F_k,F_l} in Figure 6.

The heat map of r_{F_k,F_l} (Figure 6a) is very similar to that of mutual information (Figure 5a) and learned field interaction strengths of FwFMs (Figure 5d). The corresponding Pearson correlation coefficient with mutual information is 0.5554. This implies that r_{F_k,F_l} can indeed learn different interaction strengths of different field pairs. On the other hand, the learned field interaction strengths without r_{F_k,F_l} correlates poorly with mutual information, as depicted in Figure 6b. Its Pearson correlation coefficient with mutual information is only 0.0522.

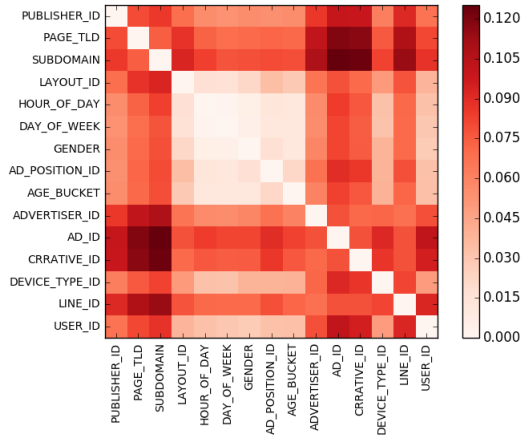
7 RELATED WORK

CTR prediction is an essential task in online display advertising. Many models have been proposed to resolve this problem such as Logistic Regression (LR) [4, 16, 21], Polynomial-2 (Poly2) [3], tree-based models [11], tensor-based models [20], Bayesian models [8], and Field-aware Factorization Machines (FFMs) [12, 13]. Recently, deep learning for CTR prediction also attracted much research attention [5, 9, 10, 17, 22, 24, 26]. However, only FFMs explicitly model different feature interactions from different fields in the multi-field categorical data, which we have shown to be very effective for CTR prediction.

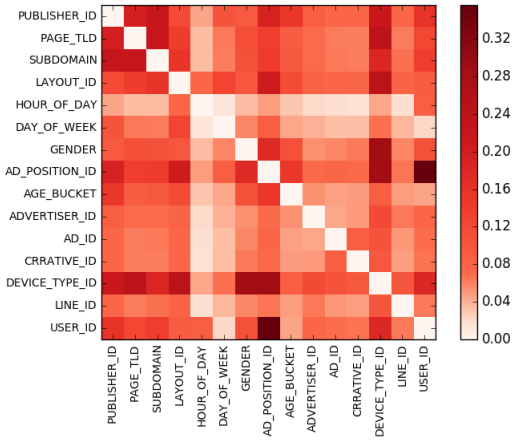
FFMs extended the ideas of Factorization Machines (FMs) [18]. The latter have been proved to be very successful in recommender systems. Recommendation is a different but very closely related topic with CTR prediction. Collaborative filtering (CF) techniques such as matrix factorization [14] have been established as standard approaches for various recommendation tasks. However, FMs have been shown to outperform matrix factorization in many recommendation tasks [12, 23] and they are also capable of effectively resolving cold-start problems [1].

8 CONCLUSION

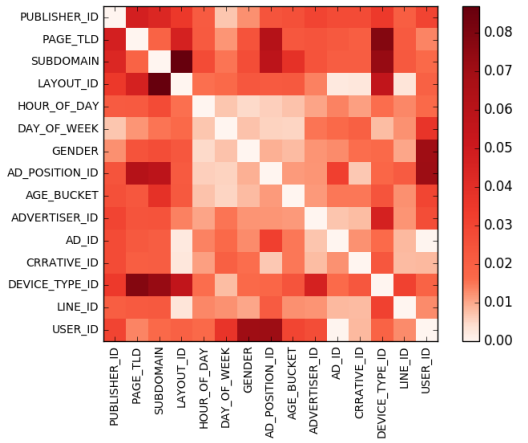
In this paper, we propose Field-weighted Factorization Machines (FwFMs) for CTR prediction in online display advertising. We show



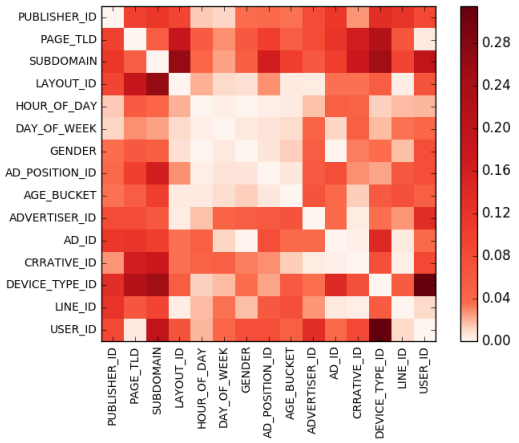
(a) Heat map of mutual informations between field pairs and labels.



(b) Heat map of learned field interaction strengths of FMs. Pearson correlation coefficient with mutual informations: -0.1210.



(c) Heat map of learned field interaction strengths of FFMs. Pearson correlation coefficient with mutual informations: 0.1544.



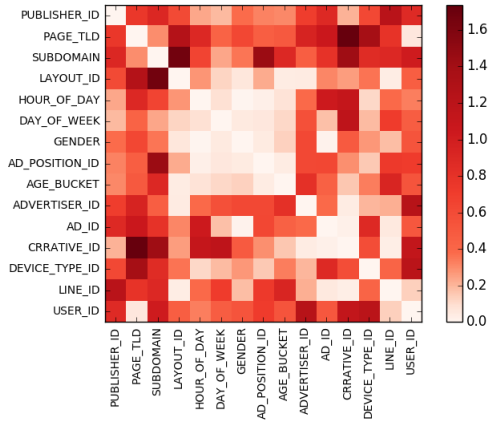
(d) Heat map of learned field interaction strengths of FwFMs. Pearson correlation coefficient with mutual informations: 0.4271.

Figure 5: Heat maps of mutual informations (a) and learned field pair interaction strengths of FMs (b), FFMs (c) and FwFMs (d) on Oath CTR data set.

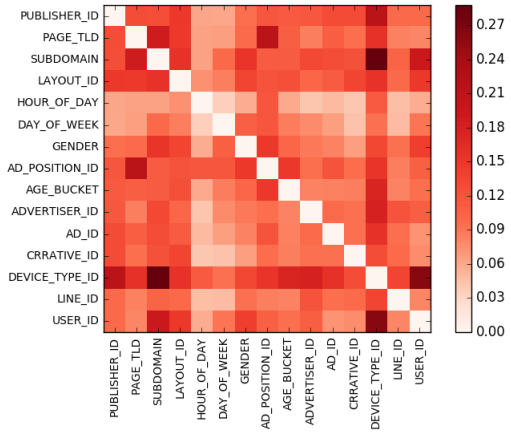
that FwFMs are competitive to FFMs with significantly less parameters. When using the same number of parameters, FwFMs can achieve consistently better performance than FFMs. We also introduce a novel linear term representation to augment FwFMs so that their performance can be further improved. Finally, comprehensive analysis on real-world data sets also verifies that FwFMs can indeed learn different feature interaction strengths from different field pairs. There are many potential directions for future research. For example, it is interesting to see whether there are other alternatives to model different field interactions rather than learning one weight for each field pair. Another direction is to explore the possibility of combining deep learning with FwFMs since the combination of deep neural networks and FMs has shown promising results in CTR prediction [9, 10, 17, 26].

REFERENCES

[1] Michal Aharon, Natalie Aizenberg, Edward Bortnikov, Ronny Lempel, Roi Adadi, Tomer Benyamini, Liron Levin, Ran Roth, and Ohad Serfaty. 2013. OFF-set: one-pass factorization of feature sets for online recommendation in persistent cold start settings. In *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 375–378.



(a) Heat map of field interaction weight r_{F_k,F_l}. Its Pearson correlation coefficient with mutual information is 0.5554.



(b) Heat map of learned field interaction strengths of FwFMs without r_{F_k,F_l}. Its Pearson correlation coefficient with mutual information is 0.0522.

Figure 6: Heat map of learned field interaction weight r_{F_k,F_l} , and heat map of learned interaction strengths between field pairs of FwFMs without r_{F_k,F_l}

[2] Interactive Advertising Bureau. 2016. IAB internet advertising revenue report. (2016). https://www.iab.com/wp-content/uploads/2016/04/IAB_Internet_Advertising_Revenue_Report_FY_2016.pdf

[3] 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.

[4] 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.

[5] 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.

[6] Thomas M Cover and Joy A Thomas. 2012. *Elements of information theory*. John Wiley & Sons.

[7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research* 9, Aug (2008), 1871–1874.

[8] 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.

[9] 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).

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

[11] 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.

[12] 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.

[13] 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.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[15] Criteo Labs. 2014. Display Advertising Challenge. (2014). https://www.kaggle.com/c/criteo-display-ad-challenge

[16] 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.

[17] 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.

[18] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

[19] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[20] 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.

[21] 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.

[22] 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.

[23] Nguyen Thai-Nghe, Lucas Drumond, Tomás Horváth, and Lars Schmidt-Thieme. 2012. Using factorization machines for student modeling.. In *UMAP Workshops*.

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

[25] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 1113–1120.

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