An Ad CTR Prediction Method Based on Feature Learning of Deep and Shallow Layers

Zai Huang[1], Zhen Pan[1], Qi Liu[1,*], Bai Long[1], Haiping Ma[2], Enhong Chen[1]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China
{huangzai,pzhen}@mail.ustc.edu.cn,{qiliuql,cheneh,blong}@ustc.edu.cn
[2]IFLYTEK Co.,Ltd., hpma@iflytek.com

ABSTRACT

In online advertising, Click-Through Rate (CTR) prediction is a crucial task, as it may benefit the ranking and pricing of online ads. To the best of our knowledge, most of the existing CTR prediction methods are shallow layer models (e.g., Logistic Regression and Factorization Machines) or deep layer models (e.g., Neural Networks). Unfortunately, the shallow layer models cannot capture or utilize high-order nonlinear features in ad data. On the other side, the deep layer models cannot satisfy the necessity of updating CTR models online efficiently due to their high computational complexity. To address the shortcomings above, in this paper, we propose a novel hybrid method based on feature learning of both Deep and Shallow Layers (DSL). In DSL, we utilize Deep Neural Network as a deep layer model trained offline to learn high-order nonlinear features and use Factorization Machines as a shallow layer model for CTR prediction. Furthermore, we also develop an online learning implementation based on DSL, i.e., onlineDSL. Extensive experiments on large-scale real-world datasets clearly validate the effectiveness of our DSL method and onlineDSL algorithm compared with several state-of-the-art baselines.

KEYWORDS

Online Advertising; CTR Prediction; Feature Learning

1 INTRODUCTION

In recent years, online advertising has been one of the most popular and effective approaches in brand promotion and product marketing [6]. As a multi-billion business, it accounts for the overwhelming majority of income for the major search engines and Internet websites. What's more, Click-Through Rate (CTR) prediction may benefit the ranking and pricing of ads and improve the income. Thus, in online advertising, CTR prediction is a crucial task, which attracts extensive concerns of both academia and industry [1, 3, 6].

In the literature, many methods have been proposed for CTR prediction, which can be mainly classified into two aspects: shallow layer models, e.g., Logistic Regression (LR) and Factorization Machines (FM), and deep layer models, e.g., Neural Networks (NN). Specifically, in shallow layer models, LR is widely applied due to its intuitiveness, extendibility and tractability [1, 6]. Furthermore, as a linear model, LR can be trained fast on large-scale datasets [1]. In contrast to LR, FM based on feature engineering and matrix design, can obtain latent relationships of each pairwise elements [8]. FM has gradually become popular in CTR prediction, as it can also well address the sparsity in ad data [7]. In deep layer models, some prior arts based on NN are developed to automatically learn high-order nonlinear features for predicting CTR more accurately [2, 5, 12]. Though both shallow layer models and deep layer models have made a great success in CTR prediction, there are some limitations of them. To be specific, there are many latent high-order nonlinear features in ad data [13], which have been proved to be beneficial to improve the performance of CTR prediction [2, 3], but shallow layer models may overlook them. On the other side, deep layer models are so computationally expensive that they cannot satisfy the necessity of updating CTR models online efficiently. Thus, how to develop a CTR prediction method, which can take full advantage of all information in ad data and be updated online fast enough at the same time, is still a challenging problem.

To solve the problem above, we propose a novel hybrid method based on feature learning of both Deep and Shallow Layers (DSL) for CTR prediction, by combining a deep layer model and a shallow layer model. Without loss of generality, in this paper, we take Deep Neural Network (DNN) as a typical deep layer model and take FM as a typical shallow layer model. Specifically, we first utilize DNN trained offline to learn the high-order nonlinear features, which can be represented as units of DNN hidden layers. Then we combine basic features and the high-order nonlinear features as the input of FM for CTR prediction. Through DSL, we can learn and make full use of basic features, pairwise interactions and high-order nonlinear features in ad data. Furthermore, we develop an online learning implementation based on DSL, i.e., onlineDSL, for updating DSL online quickly and improving

[*] Corresponding Author.

CIKM'17 , November 6–10, 2017, Singapore, Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11. . . $15.00
https://doi.org/10.1145/3132847.3133072

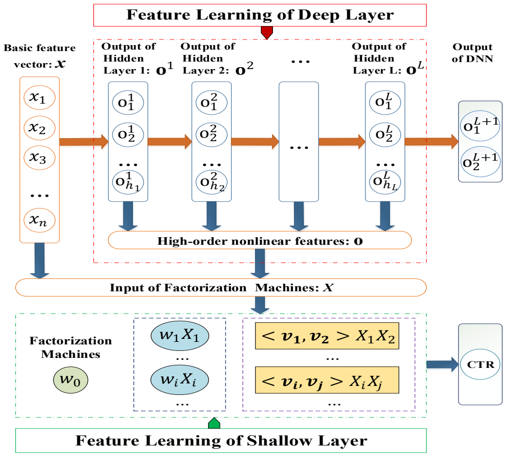

Feature Learning of Deep Layer
Basic feature vector: x
Output of Hidden Layer 1: $\mathbf{o}^1$
Output of Hidden Layer 2: $\mathbf{o}^2$
...
Output of Hidden Layer L: $\mathbf{o}^L$
Output of DNN
x_1
x_2
x_3
...
x_n
o_1^1
o_2^1
$o_{h_1}^1$
o_1^2
o_2^2
$o_{h_2}^2$
o_1^L
o_2^L
$o_{h_L}^L$
o_1^{L+1}
o_2^{L+1}
High-order nonlinear features: $\mathbf{o}$
Input of Factorization Machines: X
Factorization Machines
w_0
w_1X_1
w_iX_i
$< v_1, v_2 > X_1X_2$
$< v_i, v_j > X_iX_j$
CTR
Feature Learning of Shallow Layer

Figure 1: Framework of DSL.

the performance of CTR prediction. Finally, extensive experiments on the large-scale real-world ad datasets reveal that our proposed method can give rise to a significant improvement on CTR prediction accuracy compared with several state-of-the-art models.

2 METHODOLOGY

In this section, we first give the preliminaries of CTR prediction task. Then we introduce our proposed novel method based on feature learning of deep and shallow layers (DSL) for CTR prediction. Finally, we develop the online learning implementation based on DSL.

2.1 Preliminaries of CTR Prediction Task

We refer to a display of an ad to a particular user in a specific web as an impression. We use $\mathbf{x}$ to denote the feature vector and $y \in \{0, 1\}$ to indicate *non-click* or *click* of the impression. The CTR prediction task is to learn a function $f(\mathbf{x})$ from the impression dataset $\{(\mathbf{x}^{(i)}, y^{(i)})|i = 1, 2, \cdots, N\}$, which can be used to predict the probability an ad is clicked by a user.

2.2 DSL

The main challenges for our DSL method are in the following two aspects: how to make full use of all information in ad data for CTR prediction; how to update online efficiently. For the former challenge, in DSL, we first utilize DNN as a typical deep layer model to learn high-order nonlinear features. Then we apply FM as a typical shallow layer model for further feature learning. At last, we use the output of FM to predict CTR. For the latter one, we develop an online learning implementation based on DSL, which will be discussed in the next subsection. Figure 1 shows the framework of DSL.

2.2.1 Feature Learning of Deep Layer. The deep layer models usually have the ability to learn high-order nonlinear features automatically [5, 11]. In DNN, high-order nonlinear features can be represented as units of hidden layers [11], as shown in the red box of Figure 1. There are L hidden layers with $h_1, h_2, \cdots, h_L$ units respectively. The parameters of DNN are $(\mathbf{W}^l, \mathbf{b}^l)$, $0 < l \leq L + 1$. $\mathbf{W}^l \in R^{h_l \times h_{l-1}}$ and $\mathbf{b}^l \in R^{h_l \times 1}$ are the weight matrix and the bias vector of the l-th layer respectively. We use the vector $\mathbf{o}^l \in R^{h_l \times 1}$ to denote the output of the l-th layer units and $\mathbf{o}$ to represent the high-order nonlinear feature vector, i.e., $\mathbf{o} = \mathbf{o}^1 \| \mathbf{o}^2 \| \cdots \| \mathbf{o}^L \in R^m$, $m = h_1 + h_2 + \cdots + h_L$. "$\|$" is the operation that concatenates two vectors into a long vector. $\mathbf{o}^l$ can be expressed as

Algorithm 1 DSL for CTR prediction

```
Input: DNN parameters $(\mathbf{W}^l, \mathbf{b}^l)$, $0 < l \leq L + 1$, FM parameters $(w_0, \mathbf{w}, \mathbf{V})$, feature vector $\mathbf{x}$
Output: predicted CTR $p$
1: Compute the output $\mathbf{o}$ of all hidden layers of the DNN model with the input $\mathbf{x}$
2: Get a new vector $\mathbf{X} = \mathbf{x} \| \mathbf{o}$, $|\mathbf{X}| = |\mathbf{x}| + |\mathbf{o}|$
3: Use the FM model and the sigmoid function to predict the CTR: $p = \sigma(FM(\mathbf{X}, w_0, \mathbf{w}, \mathbf{V}))$
4: return $p$
```

$$\mathbf{o}^l = f(\mathbf{W}^l \mathbf{o}^{l-1} + \mathbf{b}^l), \tag{1}$$

where $0 < l \leq L$, $\mathbf{o}^0$ is the input of DNN, i.e., the impression feature vector $\mathbf{x}$. The dimension of $\mathbf{x}$ is n. f is an activation function. Here we use the rectified linear unit (ReLU) function $ReLU(x) = max(0, x)$ as the activation function [11].

2.2.2 Feature Learning of Shallow Layer. We use the combination of $\mathbf{x}$ and $\mathbf{o}$, i.e., $\mathbf{X} = \mathbf{x} \| \mathbf{o}$, as the input of FM. The FM model equation is the same as [8], i.e.,

$$FM(\mathbf{X}) = w_0 + \sum_{i=1}^{n+m} w_i X_i + \sum_{i=1}^{n+m} \sum_{j=i+1}^{n+m} \langle \mathbf{v}_i, \mathbf{v}_j \rangle X_i X_j, \tag{2}$$

where $w_0 \in R$, $\mathbf{w} \in R^{n+m}$ and $\mathbf{V} \in R^{(n+m) \times k}$ are the parameters of FM. X_i denotes the i-th dimension of $\mathbf{X}$. A row $\mathbf{v}_i$ within $\mathbf{V}$ is a latent vector describing X_i with k factors. k is the dimension of the latent vectors. $\langle \cdot, \cdot \rangle$ is the dot product of two vectors.

As shown in the green box of Figure 1, in the FM model, we can learn the linear relation and pairwise interactions based on the input features by the linear part $w_i X_i$ and the part $\langle \mathbf{v}_i, \mathbf{v}_j \rangle X_i X_j$ respectively.

2.2.3 CTR Prediction. As discussed above, through DSL, we can learn and make full use of basic features, pairwise interactions and high-order nonlinear features in ad data. After feature learning, utilizing the sigmoid function $\sigma(x) = \frac{1}{1 + e^{-x}}$, the CTR can be predicted as $p = \sigma(FM(\mathbf{X}))$.

Algorithm 1 shows how the DSL method predicts the CTR.

2.2.4 Parameter Learning. We first train a binary DNN model from the ad dataset with basic features. Then we train the FM model with the same dataset, using basic features and the output of all units of hidden layers as its input.

The result $\mathbf{o}^{L+1}$ of DNN is computed as Eq.(3), $|\mathbf{o}^{L+1}| = 2$. Given an ad training set with basic features, $S = \{(\mathbf{x}^{(i)}, y^{(i)})| i = 1, 2, \cdots, N, |\mathbf{x}^{(i)}| = n, y^{(i)} \in \{0, 1\}\}$, we can minimize the loss function as Eq.(4) to train the DNN parameters $(\mathbf{W}^l, \mathbf{b}^l)$, $0 < l \leq L + 1$, with the back-propagation algorithm [10].

$$o_i{}^{L+1} = softmax(\mathbf{z}^{L+1} = \mathbf{W}^{L+1} \mathbf{o}^L + \mathbf{b}^{L+1}) = \frac{e^{z_i{}^{L+1}}}{\sum_{j=1}^{2} e^{z_j{}^{L+1}}}. \tag{3}$$

$$J = -\frac{1}{N} \sum_{i=1}^{N} (y^{(i)} log(o_2^{L+1(i)}) + (1 - y^{(i)}) log(o_1^{L+1(i)})). \tag{4}$$

Algorithm 2 onlineDSL algorithm

```
Input: DNN parameters (trained offline regularly) $(\mathbf{W}^l, \mathbf{b}^l)$,
    $0 < l \leq L+1$, hyper-parameters $k, \alpha, \beta, \lambda_1, \lambda_2$
Output: FM parameters $\Theta = (w_0, \mathbf{w}, \mathbf{V})$
Initialize FTRL variables: $\forall \theta \in \Theta, z_\theta = 0, s_\theta = 0$
Initialize FM parameters $\Theta^{(0)}$ randomly
for $t = 1, \cdots, T$ do
    Receive feature vector $\mathbf{x}^{(t)}$
    Compute the output $\mathbf{o}^{(t)}$ of all hidden layers of the DNN model with the input $\mathbf{x}^{(t)}$
    Get $\mathbf{X}^{(t)} = \mathbf{x}^{(t)} \| \mathbf{o}^{(t)}$
    Predict $p_t = \sigma(FM(\mathbf{X}^{(t)}, w_0^{(t-1)}, \mathbf{w}^{(t-1)}, \mathbf{V}^{(t-1)}))$
    Observe label $y^{(t)} \in \{0, 1\}$
    Set $\mathbf{I}_\theta = \{w_0\} \cup \{w_i | w_i \in \mathbf{w}, X_i^{(t)} \neq 0\}$
        $\cup \{v_{il} | v_{il} \in \mathbf{V}, X_i^{(t)} \neq 0, 0 \leq l < k\}$
    for all $\theta \in \mathbf{I}_\theta$ do
        $g_\theta = (p_t - y^{(t)}) h_\theta(\mathbf{X}^{(t)})$, using Eq.(7)
        $\sigma_\theta = \frac{1}{\alpha}(\sqrt{s_\theta + g_\theta^2} - \sqrt{s_\theta})$
        $z_\theta = z_\theta + g_\theta - \sigma_\theta \theta^{(t-1)}$
        $s_\theta = s_\theta + g_\theta^2$
        $\delta = (\lambda_2 + \frac{\sqrt{s_\theta} + \beta}{\alpha})^{-1}$
        $\theta^{(t)} = \begin{cases} 0 & if |z_\theta| \leq \lambda_1 \\ -\delta(z_\theta - \lambda_1 sgn(z_\theta)) & otherwise \end{cases}$
    end for
end for
return $\Theta^{(T)} = (w_0^{(T)}, \mathbf{w}^{(T)}, \mathbf{V}^{(T)})$.
```

After finishing training the DNN model, for each sample $(\mathbf{x}^{(i)}, y^{(i)})$ in training set S, we can get the output $\mathbf{o}^{(i)}$ of all units of hidden layers in DNN. Then we treat the vector $\mathbf{X}^{(i)} = \mathbf{x}^{(i)} \| \mathbf{o}^{(i)}$ as the input of FM. The CTR can be predicted as $p_i = \sigma(FM(\mathbf{X}^{(i)}, w_0, \mathbf{w}, \mathbf{V}))$. We can minimize the loss function as Eq.(5) to train the FM parameters $(w_0, \mathbf{w}, \mathbf{V})$.

$$J_{FM} = -\frac{1}{N} \sum_{i=1}^{N} (y^{(i)} log(p_i) + (1 - y^{(i)}) log(1 - p_i)). \quad (5)$$

The gradient of FM parameters can be computed as Eq.(6).

$$\nabla \theta = \frac{\partial J_{FM}}{\partial \theta} = \frac{1}{N} \sum_{i=1}^{N} (p_i - y^{(i)}) h_\theta(\mathbf{X}^{(i)}). \quad (6)$$

$$h_\theta = \frac{\partial FM(\mathbf{X})}{\partial \theta} = \begin{cases} 1 & if \quad \theta = w_0, \\ X_l & if \quad \theta = w_l, \\ X_l \sum_{j \neq l}^{n+m} v_{jf} X_j & if \quad \theta = v_{lf}. \end{cases} \quad (7)$$

2.3 OnlineDSL

As various massive ad data are generated in every second, it is strongly necessary to update the CTR model online efficiently. Thus, in this subsection, we develop an online learning implementation based on DSL, i.e., onlineDSL.

Here exist two challenges in developing the onlineDSL algorithm. First, since we utilize DNN to learn high-order nonlinear features in DSL and DNN suffers heavy burden of computation, if we update DSL online directly, DNN will drag down the updating efficiency. Thus, the challenge is how to update the DNN model to ensure the performance and avoid affecting the whole updating efficiency. Second, the performance of the CTR model may be affected by the way of updating it online [6], so it is very important to find an appropriate way for updating the FM model in DSL.

For the first challenge, DNN can keep the effectiveness of learning high-order nonlinear features for a period of time after training [3, 11], so we update the DNN model offline regularly to ensure the effectiveness. For the second one, the Follow The Regularized Leader (FTRL) algorithm is more effective at not only producing a sparse solution for memory saving, but also better performance [6], so we use the FTRL algorithm for updating the FM model. To sum up, the onlineDSL algorithm takes the form of Algorithm 2.

3 EXPERIMENTS

3.1 Datasets

We use three real-world ad datasets collected from the iOS mobile terminals by IFLYTEK. Each dataset consists of training data of 7 days and test data of the subsequent day and contains ten millions of impression instances. The dimension of the feature vector of each impression is 6,070.

The statistical information of these three datasets is given in Table 1. The CTR in Table 1 is calculated as $\frac{Clicks}{Impresions}$. We name the three datasets $D1$, $D2$ and $D3$ respectively. We use $Train$ 1 and $Test$ 1 to denote the training set and the test set of $D1$. Similar names can be got for the other two datasets. We can find that all three datasets are highly unbalanced between positive instances and negative ones.

Table 1: Statistical Information of Datasets

		Period	Impressions	Clicks	CTR(%)
$D1$	$Train$ 1	20-26 May	12,737,863	246,450	1.93
	$Test$ 1	27 May	2,306,209	43,954	1.91
$D2$	$Train$ 2	25-31 May	13,669,263	228,284	1.67
	$Test$ 2	1 June	1,199,139	16,113	1.34
$D3$	$Train$ 3	28 May-3 June	10,941,412	183,668	1.68
	$Test$ 3	4 June	406,634	8,730	2.15

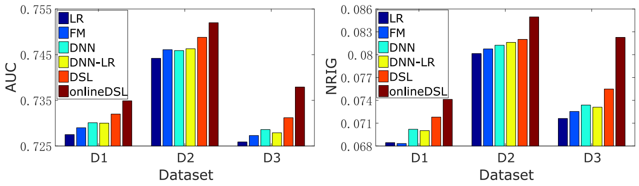
3.2 Evaluation Metrics

The performance of a CTR prediction model can be evaluated in two aspects: the accuracy of predicted click probability and the ranking quality. For them, we use normalized relative information gain (NRIG) and the area under receiver operator curve (AUC) respectively, following [4]. For NRIG and AUC, the larger, the better.

3.3 Experimental Results

3.3.1 Overall Performance. To investigate the model effectiveness, we compare the performance of our DSL method and onlineDSL algorithm with several CTR prediction models, including LR, FM, DNN and DNN-LR. DNN-LR combines DNN and LR with the basic features and the output of hidden layers of DNN as the input of LR. We set these models as baselines due to the following reasons: (1) A number of works have demonstrated that LR and FM as shallow layer models and DNN as a deep layer model are state-of-the-art models for CTR prediction [1, 6, 7, 9, 12]. (2) Since FM can capture any pairwise interaction, we use FM instead of LR as the shallow layer model in DSL.

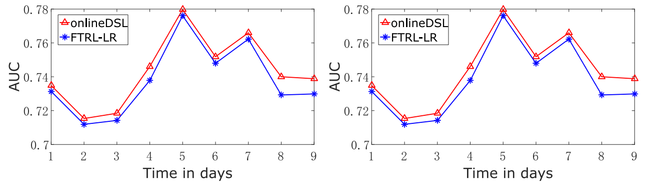
In experiments, we set the number of hidden layers $L = 3$, the dimension of latent vector $k = 8$, and hyper-parameters of FTRL $\alpha = 0.03$, $\beta = 0.5$, $\lambda_1 = 0.001$, $\lambda_2 = 0.05$.



(a) AUC

(b) NRIG

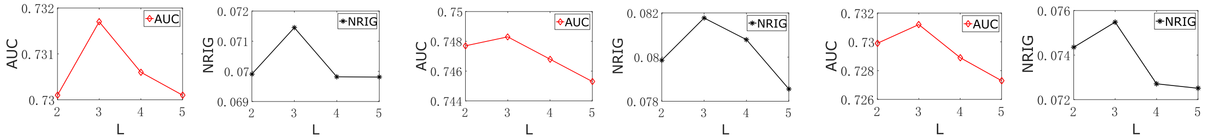
Figure 2: Performance comparison.



(a) AUC

(b) NRIG

Figure 3: Performance of onlineDSL and FTRL-LR over time.



(a) $D1$

(b) $D2$

(c) $D3$

Figure 4: The influence of L on DSL.

Figure 2 shows the performance results of all the models. It can be observed that DSL and onlineDSL significantly outperform the baselines in all three datasets on NRIG and AUC. Specifically, onlineDSL is much better than DSL because updating online can enhance the model effectiveness. These results suggest that DSL and onlineDSL can more effectively predict CTR by making full use of basic features, pairwise interactions and high-order nonlinear features.

3.3.2 Effectiveness of onlineDSL Over Time. We validate the effectiveness of onlineDSL as time goes on, comparing it with FTRL-LR. FTRL-LR is an online algorithm for CTR prediction by combining FTRL and LR, and has achieved a great success in industry [6]. As DNN in DSL need be trained on training data, we use $Train$ 1, 7-days data (20 May - 26 May) as the training set and the union of $Test$ 1 and $D3$, 9-days data (27 May - 04 June), as the test set. In the test phase, the DNN model in onlineDSL is updated daily.

Figure 3 shows the performance comparison of onlineDSL and FTRL-LR over time. It is easy to find that onlineDSL brings a significant performance improvement on both NRIG and AUC, compared with FTRL-LR. This result indicates the effectiveness of onlineDSL for CTR prediction.

3.3.3 Sensitivity of Hyper-Parameter. According to the setting of experiments, there is a key hyper-parameter having the most influence on the performance of DSL, i.e., the number of hidden layers L.

We conduct an experiment for different L from the set $\{2, 3, 4, 5\}$ and the results are shown in Figure 4. We can find that DSL can get the best performance when $L = 3$ and both AUC and NRIG decrease when $L > 3$ or $L < 3$. We guess a possible reason is that the complexity of the DNN model is appropriate to capture high-order nonlinear features effectively when $L = 3$, but the DNN model easily becomes overfitting when $L > 3$ and underfitting when $L < 3$.

4 CONCLUSION

In this paper, we proposed a novel DSL method for CTR prediction, which could make full use of basic features, pairwise interactions and high-order nonlinear features in ad data. Meanwhile, we developed the onlineDSL algorithm for updating DSL online. Finally, extensive experimental results on real-world ad datasets demonstrated the effectiveness of our DSL method and onlineDSL. We should note that, DSL is actually a general method. In the future, we will test its performance with other deep layer models (e.g., Autoencoder) and shallow layer models (e.g., Support Vector Machine).

5 ACKNOWLEDGEMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2016YFB1000904), the National Natural Science Foundation of China (Grants No. 61672483 and U1605251), and the Youth Innovation Promotion Association of CAS (No. 2014299).

REFERENCES

[1] O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *TIST*, 5(4):61, 2015.

[2] J. Chen, B. Sun, H. Li, H. Lu, and X.-S. Hua. Deep ctr prediction in display advertising. In *MM*, pages 811–820. ACM, 2016.

[3] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *ADKDD*, pages 1–9. ACM, 2014.

[4] C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey. Click-through prediction for advertising in twitter timeline. In *KDD*, pages 1959–1968. ACM, 2015.

[5] Q. Liu, F. Yu, S. Wu, and L. Wang. A convolutional click prediction model. In *CIKM*, pages 1743–1746. ACM, 2015.

[6] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad click prediction: a view from the trenches. In *KDD*, pages 1222–1230. ACM, 2013.

[7] Z. Pan, E. Chen, Q. Liu, T. Xu, H. Ma, and H. Lin. Sparse factorization machines for click-through rate prediction. In *ICDM*, pages 400–409. IEEE, 2016.

[8] S. Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.

[9] S. Shang, L. Chen, C. S. Jensen, J.-R. Wen, and P. Kalnis. Searching trajectories by regions of interest. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1549–1562, 2017.

[10] D. Williams and G. Hinton. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

[11] D. Yu and L. Deng. *Automatic speech recognition: A deep learning approach*. Springer, 2014.

[12] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, 2014.

[13] Z. Zhao, X. He, D. Cai, L. Zhang, W. Ng, and Y. Zhuang. Graph regularized feature selection with data reconstruction. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):689–700, 2016.