A Convolutional Click Prediction Model

Qiang Liu, Feng Yu*, Shu Wu, Liang Wang
Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{qiang.liu, feng.yu, shu.wu, wangliang}@nlpr.ia.ac.cn

ABSTRACT

The explosion in online advertisement urges to better estimate the click prediction of ads. For click prediction on single ad impression, we have access to pairwise relevance among elements in an impression, but not to global interaction among key features of elements. Moreover, the existing method on sequential click prediction treats propagation unchangeable for different time intervals. In this work, we propose a novel model, Convolutional Click Prediction Model (CCPM), based on convolution neural network. CCPM can extract local-global key features from an input instance with varied elements, which can be implemented for not only single ad impression but also sequential ad impression. Experiment results on two public large-scale datasets indicate that CCPM is effective on click prediction.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Filtering

Keywords

Click Prediction, Convolution Neural Network

1. INTRODUCTION

Recently, online advertising has become the most popular approach to do brand promotion and product marketing for the *advertiser*, and contributes the overwhelming majority of income for the commercial web *publisher*.

Nowadays, click prediction on single ad impression [10] has received much attention, and many different approaches have been proposed. For simplicity and effectiveness, Logistic Regression (LR) [7, 9] has been widely used in click prediction. Representing each *element* (e.g. query, ad, user and other contexts) of a single ad impression by a value, LR is not capable enough to describe the latent features of an element or reveal the complicated relation among these elements. As a widely-used technique in recommendation systems, matrix factorization (MF) method [4] in the Collaborative Filtering approach is also employed for click prediction. MF method factorizes and rebuilds the dependency matrix to learn latent semantic representations of pages and ads. Later, Factorization Machines (FM) [5, 6], a extension of MF in multiple element space, obtains latent semantic information of each pairwise elements, which is able to better model relation of various elements. However, MF and FM models capture relevance of pairwise elements in single ad impression and overlook the high-order interaction among these elements.

Different from traditional works taking single ad impression as input instance and overlooking dependency of historical impressions, Recurrent Neural Network (RNN) model [10] is leveraged for click prediction of sequential ad impression. Taking full advantage of historical click sequences, the recurrent structure enhances the accuracy of click prediction further. The model takes each user's browsing history as a sequence and obtains internal sequential dependency of varied impressions. Historical click sequence of a certain user is divided by different time intervals, sequence signals of one time interval can be propagated to next interval by the recurrent connection matrix. Due to the fact that the recurrent connection matrix of a trained RNN model is a constant one, the propagations of sequence signals between every two consecutive time intervals remain all the same. However, in real-world scenarios, since users' attitudes toward ads change over time, RNN models may has its limitation for these scenarios due to using the unchangeable propagations.

In order to mine significant semantic features in complex and dynamic sceneries, deep neural network is a good choice. As stated above, for click prediction on single ad impression, the MF and FM methods only reveal the relevance between pairwise elements, but convolutional neural network (CNN) can treat varied elements in a single ad impression as a whole and obtain complex interaction among them. On the other hand, the unchangeable propagations of RNN models on sequential ad impression has the limitation in effectively modeling dynamic click predictions, while pooling and convolutional layers of a deep CNN architecture can fully extract local-global key features from sequential ad impression. In addition, some recent studies about CNN architecture have successfully model significant semantic features in varieties of fields. CNN approaches to speech recognition [1], image recognition [3], information retrieval [8] have achieved much improvement in respective fields. Moreover, proved

*This author contributed equally as the first author.

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806603.

as a effective sentence model in natural language processing, Dynamic Convolutional Neural Network (DCNN) [2] can analyses semantic content and extracts key features of sentences.

We propose a Convolutional Click Prediction Model (CCPM) for click prediction in sceneries of the single ad impression and sequential ad impression. An input instance of CCPM is composed by elements of an ad impression or elements related to a sequential ad impression. Convolutional layers extract local-global features of input instances, and the dynamic pooling layers can obtain significant features. CCPM investigates significant semantic features of an ad impression and sequential relevance of impression history into enhancing the accuracy of click prediction. Experiments are conducted to validate the CCPM model's effectiveness in modeling different kinds of input instances and reveal that CCPM achieves great improvement on the accuracy of click prediction comparing the state-of-the-art models such as LR, FM and RNN. To the best of our knowledge, CCPM is the first approach that attempts to leverage CNN to improve the accuracy of click prediction.

2. CCPM

In an event of *single ad impression*, there are some noticeable elements like user, query, ad, impression time, site category, device type, etc. On the other hand, sometimes system can collect sequential ad impression of each individual user, where user's behaviors on ads yield high dependency on how the user behaved along with the past time. This *sequential ad impression* is comprised of a series of single ad impressions. The goal of this work is to predict the click probability based on these two kinds of impressions.

We model above input instances using a convolutional architecture that alternates wide convolutional layers with flexible p-max pooling layers. The whole procedure of CCPM is illustrated in Figure 1. In the network the width of an intermediate feature map varies with the length of the input instance. It is remarkable to state that the proposed model can handle input instances with varied length, which make it can be used widely.

2.1 Convolution Layer

Given an input instance with n elements, to obtain the first layer of CCPM, we take an embedding $\mathbf{e}_i \in R^d$ for each element in the instance and construct the instance matrix $\mathbf{s} \in R^{d \times n}$ as

$$\mathbf{s} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{e}_1 & \cdots & \mathbf{e}_n \\ \vdots & \vdots & \vdots \end{bmatrix} . \tag{1}$$

The values in the embeddings $\mathbf{e}_i$ are estimated during the training process, which contributes to more suitable representations for input instances. A convolutional layer in the network is obtained by convolving a weight matrix $\mathbf{w} \in R^{d \times \omega}$ with the activation matrix at the layer below in an one-dimensional row-wise way. For example, the second layer is obtained by applying a convolution on the input instance matrix $\mathbf{s}$. Dimension d and filter width ω are hyper-parameters of input instances. The resulting matrix $\mathbf{r}$ has dimensions $d \times (n + \omega - 1)$. Given $\mathbf{w}_i \in R^{\omega}$, $\mathbf{s}_i \in R^n$ and $\mathbf{r}_i \in R^{(n+\omega-1)}$ as the i-th row of corresponding matrix, we can obtain one-dimensional convolution as

$$\mathbf{r}_i = {\mathbf{w}_i}^T \mathbf{s}_{i,j-\omega+1:j} , \tag{2}$$

where the index j ranges from 1 to $n + \omega - 1$. Out-of-range values $\mathbf{s}_{i,k}$ (where $k < 1$ or $k > n$) are set to zero.

The optimized weights in the filter $\mathbf{w}$ detects features and recognizes specific ranges of neighborhood in input instances. Applying one-dimensional row-wise convolution on two-dimensional matrix of activations, has the following advantage over simply using two-dimensional convolution. Usually we apply two-dimensional convolution in image identification for the reason that the detectors need to recognize special two-dimensional features, such as edges of an objective. However, in the click prediction model, each dimension of the embedding represents a distinct aspect of an element in an instance. Therefore, each row of the resulting matrix $\mathbf{r}$ obtains distinct features from the activation matrix.

2.2 Flexible p-Max Pooling

Here, we describe the flexible p-max pooling layer. Given a vector $\mathbf{r}_i \in R^n$, p-max pooling selects a sub-vector $\mathbf{s}_i^p \in R^p$, which contains the p biggest values in the original vector $\mathbf{r}_i$. Due to the fact that input instances are of varied length, the vector lengths of intermediate convolutional layer change accordingly, consequently the following pooling layer need to be flexible enough to select prominent features smoothly. Considering all facts mentioned above, we let p be a function of length of the input instance and depth of the network. In spite of many possible functions, we select the following one

$$p_i = \begin{cases} (1 - (i/l)^{l-i})n, & i = 1, \cdots, l-1 \\ 3, & i = l \end{cases} , \tag{3}$$

where l is the total number of convolutional layers of the network, n is the length of the input instance and p_i represents the parameter of the i-th pooling layer. For example, given an input instance of length $n = 18$, in a network of three convolutional layers, whose pooling parameters are as follows: $p_1 = 16$, $p_2 = 6$ and $p_3 = 3$.

This selected function has many advantages. Firstly, the last pooling layer has a fixed parameter, so it is guaranteed that the matrix of the fully connected layer for output has a unified dimensionality, despite varied lengths of different input instances. Secondly, the power-exponential function changes slowly at first compared with linear function, which avoids losing too many important features at the beginning.

The flexible p-max pooling layer can not only select the p most key features, but also preserve the relative order of those features, which plays a critical role in the sequential click prediction.

2.3 Feature Maps

We apply a non-linear function for outputs of pooling layers. The non-linear function is also called as activation function, which obtains activations of threshold values:

$$\tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} . \tag{4}$$

So far, convolutional layer, flexible p-max pooling layer and non-linear function have been applied to input instances. In this way, we can obtain a first order feature map. Moreover, the three operations above can be repeated again and again to yield multiple order feature maps and a architecture of

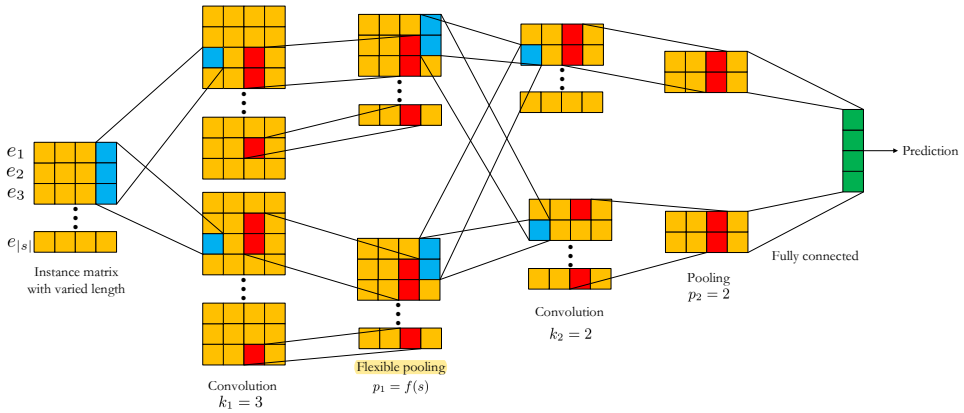

Figure 1: The framework of CCPM. The left part is an input instance (single ad impression or sequential ad impression) with varied elements and the length of element embedding is $d = 4$. The architecture has two convolutional layers with two feature maps each. The widths of filters at two layers are three and two respectively. The flexible pooling layer p_1 changes with length of instance and the last pooling layer $p_2 = 2$.

deeper layers. We denote an i-th order feature map by $\mathbf{F}^i$. At a certain layer, many feature maps can be computed in parallel. For example, $\mathbf{F}^i_j$ represents the j-th feature map of those i-th order feature maps, an is computed by summing the convolutional results of a distinct weight matrix $\mathbf{w}^i_{j,k}$ and each feature map $\mathbf{F}^{i-1}_k$ of the lower order $i-1$,

$$\mathbf{F}^i_j = \sum_{k=1}^{m_i} \mathbf{w}^i_{j,k} * \mathbf{F}^{i-1}_k, \tag{5}$$

where m_i denotes the number of feature map in corresponding i-th order layer, and $*$ refers to the one-dimensional row-wise convolution described in Sec. 2.1. Similarly, flexible p-max pooling and non-linear function can be applied to feature map $\mathbf{F}^i_j$ successively. Finally, there is a fully connected layer, and the prediction is made via softmax.

3. EXPERIMENTS

3.1 Datasets and Baselines

To empirically evaluate the performance of our method on the click prediction with single and sequential impression data, we perform experiments on two public real-world datasets: Avazu[1] and Yoochoose[2]. The Avazu dataset includes several days of ad click-through data, ordered chronologically. In each piece of click data, there are 17 data fields such as ad id, site id, click, etc. These above data fields indicate elements of a single ad impression. We use this dataset to assess the performance of click prediction on the single ad impression. Collected during several months in 2014, the Yoochoose dataset contains many sessions of browse and purchase events from an online retailer, where each session encapsulates the click events of an individual user. Some sessions contain purchase events, which means that the session ends with the user purchasing something. Here, we treat products as ads, then the browse behavior can be viewed as a single ad impression and the purchase behavior as an impression with click. This dataset is employed to evaluate the performance of click prediction on the sequential ad impression.

[1]https://www.kaggle.com/c/avazu-ctr-prediction/data
[2]http://recsys.yoochoose.net

Three state-of-the-art methods are used for empirical comparison, which are LR [7], FM [6] and RNN [10]. (1) As a widely used algorithm for click prediction in industry, LR is easy to understand, quick to train, and efficient enough to be implemented by search engines as an integral part of their advertising system. (2) FM is a general regression model that captures interaction between pairs of elements by using factors. FM has proved to be useful in different tasks and domains. In particular, it can be efficiently used to model the interaction with various elements of ad impressions. (3) RNN models the dependency on user's sequential behaviors into the click prediction process, which depends on not only the current input features, but also the sequential historical information. Since Avazu does not contain sequential ad impression, we only implement RNN model on Yoochoose. In all experiments, we randomly select 90% of dataset as training data and the rest 10% as test data. For CCPM, we apply a CNN architecture of three layers in this work. The parameters of CCPM are set as d=11, m=[4,4,2], w=[6,5,3] for the Avazu dataset, and d=8, m=[3,4,2], w=[6,5,3] for Yoochoose (m, w are the number of feature maps and filter width in three layers).

In the real-world scenarios, the probability of click is extremely low, similar to [7], we adopt logloss as the evaluation metric to measure the accuracy of CTR prediction.

$$\log loss = -\frac{1}{n_{test}} \left[\sum_{i=1}^{n_{test}} y_i \log p_i + (1 - y_i) \log(1 - p_i) \right], \tag{6}$$

where $p_i = P(y_i = 1|\mathbf{s})$ represents the predicted click probability. and $\mathbf{s}$ denotes an ad impression. y_i is the corresponding observed label, $y_i = 1$ means the user has click the ad impression. m is the total number of input instances.

3.2 Results and Analyses

Left part of Figure 2 illustrates the click prediction performance of CCPM and other competitive compared methods on single ad impression and sequential ad impression. We identify that on both datasets, CCPM outperfom the conventional methods. Since FM can describe the latent features of an element and reveal the relation of pairwise elements, it achieves significant improvement over the that of LR on both datasets. On sequential ad impression, RNN

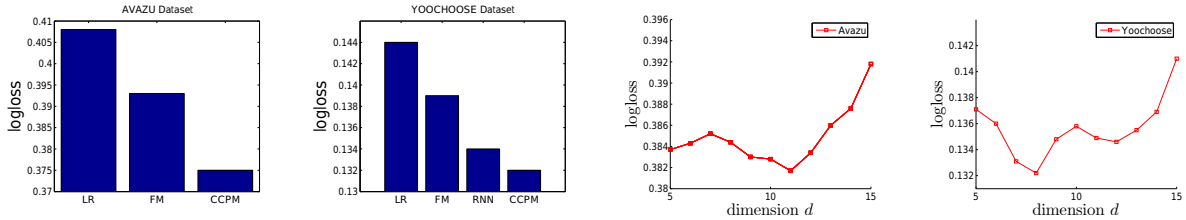

Figure 2: Experimental results (left) and the impact of dimensionality (right) of CCPM on the Avazu dataset and the Yoochoose dataset.

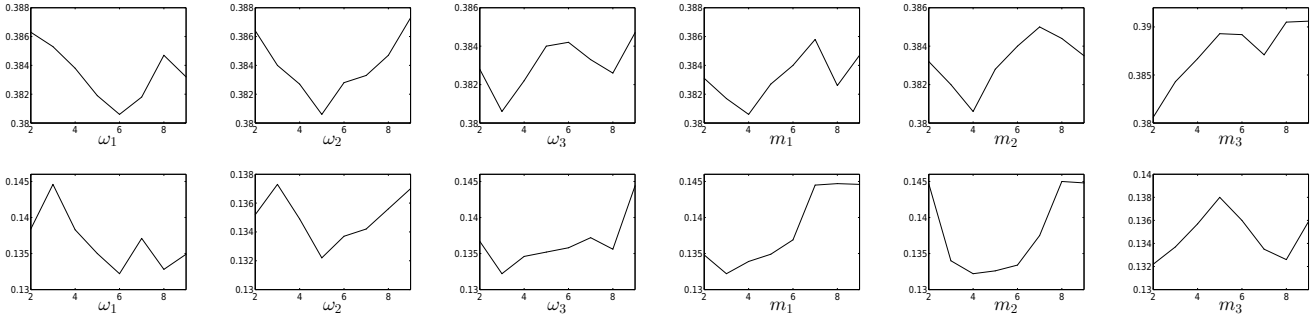

Figure 3: Parameter Study of filter width w and the number of feature maps m in corresponding layer measured by LogLoss. The top part illustrates the results on the Avazu dataset, and the bottom part illustrates the results on the Yoochoose dataset.

leverages sequential dependency of varied impressions, and enhance the effectiveness of click prediction further. Since CCPM obtain underlying semantic information of input instances and extracts local-global features by using convolutional layers, and use k-max pooling to select key features, it can not only reveal the high-order interaction among various elements of a single ad impression but also capture the historical propagation pattern in sequential ad impression.

Furthermore, in the right part of Figure 2, we illustrate the logloss values of CCPM on both datasets with varying dimensionality d of latent vector. On the Avazu dataset, the performance of CCPM achieves the best result at $d = 11$, while on Yoochoose CCPM yields the best performance when the dimensionality $d = 6$. It may be because the Yoochoose dataset is more sparse than the Avazu, latent vector with small dimension can be well estimated. After CCPM obtains the best results on both datasets, the performance decreases gradually with increasing d due to overfitting.

Finally, on both datasets, the parameter impacts of the filter width w and the number of feature map m in corresponding layer are studied. As illustrated in Figure 3, setting smaller corresponding filter width in deeper convolutional layer will contribute to higher accuracy of click prediction. The filters w of convolution layers can learn to recognize specific neighborhoods that have size less or equal to the filter width w. Therefore as reflected in the experiment results, w_1 in the first layer is often set to large enough to grasp all possible neighborhoods. Considering that pooling layers will drop some less significant items, input length of following convolutional layer can decreases. As a result, key features of input instances are further extracted at a deeper layer and kernel size gets smaller. For the sake of enriching the representation of input instances from various angles, there are many parallel feature maps in one layer. Similarly, we can also set smaller number of feature maps in deeper layer to reach better click prediction results. As a layer goes deeper, key features have already been extracted and noise eliminated, deeper layers just need small number of feature maps to extract key features.

4. CONCLUSIONS

In this paper, we have proposed a convolutional click prediction model based on CNN for single and sequential ad impression. Extensive experiments on two public datasets have demonstrated the effectiveness of the proposed model.

5. ACKNOWLEDGMENTS

This work is jointly supported by National Basic Research Program of China(2012CB316300), and National Natural Science Foundation of China (61403390, U1435221, 61175003, 61420106015).

6. REFERENCES

[1] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *ICASSP*, 2012.

[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *ACL*, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[4] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *KDD*, 2011.

[5] S. Rendle. Factorization machines with libfm. *ACM TIST*, 3(3):57, 2012.

[6] S. Rendle. Social network and click-through prediction with factorization machines. In *KDD Cup Workshop*, 2012.

[7] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *WWW*, 2007.

[8] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *CIKM*, 2014.

[9] L. Yan, W.-j. Li, G.-R. Xue, and D. Han. Coupled group lasso for web-scale ctr prediction in display advertising. In *ICML*, 2014.

[10] 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.