Translation Method of Contextual Information into Textual Space of Advertisements

Yukihiro Tagami
Yahoo Japan Corporation
Tokyo, Japan
yutagami@yahoo-corp.jp

Toru Hotta
Yahoo Japan Corporation
Tokyo, Japan
thotta@yahoo-corp.jp

Yusuke Tanaka
Yahoo Japan Corporation
Tokyo, Japan
yuustana@yahoo-corp.jp

Shingo Ono
Yahoo Japan Corporation
Tokyo, Japan
shiono@yahoo-corp.jp

Koji Tsukamoto
Yahoo Japan Corporation
Tokyo, Japan
kotsukam@yahoo-corp.jp

Akira Tajima
Yahoo Japan Corporation
Tokyo, Japan
atajima@yahoo-corp.jp

ABSTRACT

Contextual advertising has a key problem to determine how to select the ads that are relevant to the page content and/or the user information. We introduce a translation method that learns a mapping of contextual information to the textual features of ads by using past click data. This method is easy to implement and there is no need to modify an ordinary ad retrieval system because the contextual feature vector is simply transformed into a term vector with the learned matrix. We applied our approach with a real ad serving system and compared the online performance in A/B testing.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Commercial services*; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Contextual advertising, Learning-to-rank, Click feedback.

1. INTRODUCTION

Contextual advertising is a form of textual advertising usually displayed on third party web pages. The advertiser is primarily interested in targeting relevant users, and the publisher is concerned about keeping the user experience pleasant. To satisfy these two objectives, an ad-networking service selects ads that are relevant to the page content and/or the user information.

The relevance of an ad to page content is typically a tf-idf score that measures the word overlap between the page content and ad content, but this is not very effective if the vocabulary used in the page is expected to be different from the vocabulary used in the ad. To remedy this problem, some previous studies used a semantic taxonomy [1] or hidden classes [2]. However, in these approaches, it is necessary to expand the ad retrieval system to handle the categories or classes.

We introduce here an approach that does not require modification of an ordinary ad retrieval system, which calculates a matching score between two term vectors. This approach is a method of translating ad request information into the textual space of ads. With this translation table, the feature vector of ad requests is transformed into the input term vector of the ad retrieval system.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577275.

2. METHODS

We define a score proportional to click-through rate (CTR) for query feature vector $\boldsymbol{q} = (q_1, \ldots, q_{D_q})^{\mathrm{T}}$ and ad feature vector $\boldsymbol{a} = (a_1, \ldots, a_{D_a})^{\mathrm{T}}$ as follows:

$$score(\boldsymbol{q}, \boldsymbol{a}) = bscore(\boldsymbol{q}, \boldsymbol{a}) + tscore(\boldsymbol{q}, \boldsymbol{a}). \quad (1)$$

$\boldsymbol{q}$ is a query feature vector of ad request, which includes web page and user information. $bscore(\boldsymbol{q}, \boldsymbol{a})$ is a basic score as $bscore(\boldsymbol{q}, \boldsymbol{a}) = \boldsymbol{w}_{basic}^{\mathrm{T}} \boldsymbol{x}_{basic}$. $\boldsymbol{x}_{basic}$ is a feature vector which includes features such as the ad's own clickability and similarity scores like term vector cosine. $\boldsymbol{w}_{basic}$ is a weight vector corresponding to $\boldsymbol{x}_{basic}$. We also define a matching score $tscore(\boldsymbol{q}, \boldsymbol{a})$ using translation matrix $\boldsymbol{W} = [w_{ij}]_{D_q \times D_a}$ as follows:

$$tscore(\boldsymbol{q}, \boldsymbol{a}) = \boldsymbol{q}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{a} = \sum_{i=1}^{D_q} \sum_{j=1}^{D_a} w_{ij} q_i a_j.$$

In our approach, we transform $\boldsymbol{q}$ with $\boldsymbol{W}$ and use this for ad retrieval. Therefore, we need to learn the matrix. The reason for adding $bscore(\boldsymbol{q}, \boldsymbol{a})$ is that the score proportional to CTR consists not only of a interaction between the query and ad but other factors also such as $\boldsymbol{x}_{basic}$.

For computational efficiency in learning the matrix and retrieving the ads, we first select ad features related to each query feature and then learn the corresponding w_{ij}, instead of the approach which learns directly with a large hash table and L1 regularization [4].

We calculate a score $m_{ij} = \frac{ctr(q_i, a_j)}{ctr(a_j)}$ for each pair of q_i and a_j presented in the training data. $ctr(a_j)$ denotes the CTR of ads that include feature a_j. Similarly, $ctr(q_i, a_j)$ represents the CTR of ads that include feature a_j when query feature vector includes q_i. So, large m_{ij} value means that ads that include feature a_j are more likely to be clicked when query feature vector includes q_i. For each q_i, some of the a_j that have larger m_{ij} are selected:

$$A_i = \{j \mid \text{where } m_{ij} \text{ in the top } M_{filter} \text{ for } i\}$$

where M_{filter} is a hyper-parameter. The number of nonzero elements in $\boldsymbol{W}$ increases as a function of M_{filter}. We use

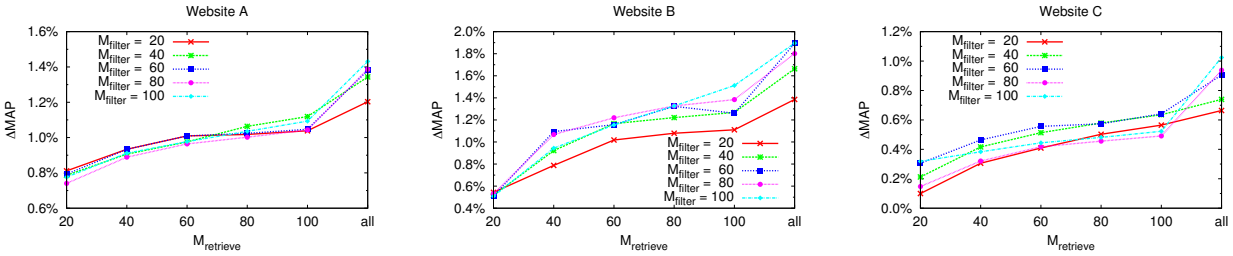

Figure 1: Offline experimental results

A_j and replace the score (1) as follows:

$$\begin{aligned} score(\boldsymbol{q}, \boldsymbol{a}) &= \sum_{i=1}^{D_q} \sum_{j \in A_i} w_{ij} q_i a_j + \boldsymbol{w}_{basic}^{\mathrm{T}} \boldsymbol{x}_{basic} \\ &= \boldsymbol{w}_{trans}^{\mathrm{T}} \boldsymbol{x}_{trans} + \boldsymbol{w}_{basic}^{\mathrm{T}} \boldsymbol{x}_{basic} \\ &= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}. \end{aligned}$$

As described in our previous work [3], we define a pairwise loss function like RankSVM for **clicked requests**, which is a set of ad requests that include at least one clicked impression. So we learn the $\boldsymbol{w}$ by using past click data.

With the learned matrix $\mathbf{W}$, the query feature vector is transformed into the input term vector of the ad retrieval system for each ad request: $\boldsymbol{q}_{input} = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{q}$. We need to limit the number of nonzero values in the input vector because the performance of the ad retrieval system declines in accordance with the number of these values. In this paper, we simply choose top-$M_{retrieve}$ elements, which are larger values.

3. EXPERIMENT

This section describes offline and online evaluations. Due to business confidentiality, we report only relevant performance when showing experimental results.

We compare the models using the data sampled from an ad network for a period of six weeks. The models we evaluate are constructed with respect to each website, since the web page and the users to visit there are different.

The query features $\boldsymbol{q}$ include web page and user information. The web page features are extracted terms which are scored with their position in the page and HTML tags. The user features are terms and categories which the user is interested in, as well as gender, age, and location. In this paper, we simply use textual features as the ad features $\boldsymbol{a}$, which are tf-idf weighted terms based on the title and description. The basic features $\boldsymbol{x}_{basic}$ include the ad features, the display position on the web page, and some common similarity values such as term vector cosine. The ad's own features include the tf-idf terms and the historical CTR of the ad and advertiser.

Offline evaluation. We compared the proposed method with a baseline model that only uses $\boldsymbol{x}_{basic}$ instead of $\boldsymbol{x} = (\boldsymbol{x}_{trans}^{\mathrm{T}}, \boldsymbol{x}_{basic}^{\mathrm{T}})^{\mathrm{T}}$. As described in the above, we need to limit the number of query terms by reason of the performance of the ad retrieval system. In this offline experiment, we carried out our evaluation by changing the value of $M_{retrieve}$. We changes $M_{retrieve}$ and truncate the query term vector during the evaluation, not during training. We evaluated the performance of the model by using the mean average precision (MAP) and normalize the scores of the method by the above baseline model. We investigated model performance of each M_{filter} when changing $M_{retrieve}$. The experimental results are reported in Figure 1. Our model achieved improvement over the baseline model in all websites. However, trends of the results are different from each website. This results indicate that optimal M_{filter} varies depending on both $M_{retrieve}$ and website.

Online evaluation. To measure the online performance, we applied our approach to a real ad serving system. This ad serving system adopts a two-stage approach. The first stage retrieves K ads in total from an ad corpus by multiple methods. The second stage selects the desired top-k using brute force CTR prediction on the K retrieved ads ($k \ll K$). We added the proposed method to the first stage and compared the online performance by conducting A/B testing. The online test ran over five days period in August 2013 for website A and B. We chose these two websites, since the improvement of both websites are fairly large in the offline evaluation. Hyper-parameters are set as follows: ($M_{filter} = 60, M_{retrieve} = 20$).

The experimental results are summarized in Tables 1. This result indicates that our proposed method achieved improvement in the online setting as well as offline setting. In website A, CPC greatly increased, regardless of a slight CTR lift. One possible explanation for this result is that ads were ranked by considering revenue in this online testing. Thus, CPC increased instead of CTR in website A.

Table 1: Online A/B testing results. Values represent the relative gains.

Metric	Website	
	A	B
Click-through rate (CTR)	+0.06%	+4.68%
Cost per click (CPC)	+12.89%	+1.73%
Revenue per request (RPR)	+12.96%	+6.49%

4. REFERENCES

[1] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *SIGIR*, 2007.

[2] A. Ratnaparkhi. A hidden class page-ad probability model for contextual advertising. In *Workshop on Targeting and Ranking for Online Advertising at the 17th International World Wide Web Conference*, 2008.

[3] Y. Tagami, S. Ono, K. Yamamoto, K. Tsukamoto, and A. Tajima. Ctr prediction for contextual advertising: learning-to-rank approach. In *ADKDD*, 2013.

[4] C. Wang, R. Raina, D. Fong, D. Zhou, J. Han, and G. Badros. Learning relevance from heterogeneous social network and its application in online targeting. In *SIGIR*, 2011.