CTR Prediction for Contextual Advertising: Learning-to-Rank Approach

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

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

Koji Yamamoto
Yahoo Japan Corporation
Tokyo, Japan
koyamamo@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 is a textual advertising displayed within the content of a generic web page. Predicting the probability that users will click on ads plays a crucial role in contextual advertising because it influences ranking, filtering, placement, and pricing of ads. In this paper, we introduce a click-through rate prediction algorithm based on the learning-to-rank approach. Focusing on the fact that some of the past click data are noisy and ads are ranked as lists, we build a ranking model by using partial click logs and then a regression model on it. We evaluated this approach offline on a data set based on logs from an ad network. Our method is observed to achieve better results than other baselines in our three metrics.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation.

Keywords

Click-through rate prediction, Contextual advertising, Learning-to-rank.

1. INTRODUCTION

Contextual advertising is a textual advertising usually displayed on third party web pages. In the common pay-per-click model, the advertiser pays the web publisher a fee only if a user clicks their advertisements and visits the advertiser's site. To maximize revenue by choosing some ads from candidates and ranking them to display, an advertising system needs to predict the expected revenue for each ad. The expected revenue from displaying each ad is a function of both bid price and click-through rate (CTR). Bid price is the cost that the advertiser agrees to pay per click, so the advertising system already knows this. On the other hand, CTR for each ad can vary significantly in accordance with many factors such as the web page and user, so the system then needs to predict CTR for each ad. Note that CTR prediction is not only related to web publisher's revenue but user's experience and advertiser's payment because this influences ranking, filtering, placement, and pricing of ads. This is why CTR prediction plays an important role in contextual advertising.

CTR prediction for each ad is typically based on a statistical model trained with past click data. Examples of such statistical models are logistic regression [9, 10], probit regression [15] and boosted trees [12, 27].

In contextual advertising, CTR generally ranges from 0.001% to 0.1%, which is very low with high variance [4]. Thus, the training data are highly skewed towards the non-clicked class.

In this paper, we focus on some of the ad requests by web publishers. Typically, web publishers request some ads simultaneously because some ads are displayed as lists on one page at a time. In other words, one ad request includes some impressions of ads. There are two kinds of ad requests: **clicked requests** and **non-clicked requests**. Any ad request in clicked requests includes at least one clicked impression of the ad. On the other hand, non-clicked requests do not include clicked impressions. Figure 1 shows examples of clicked requests and non-clicked requests.

In clicked requests, assuming the ad was not clicked by mistake, the user is aware of the clicked ad and the neighboring ads. Naturally, the clicked ad is a good positive sample. In addition, these neighboring ads are considered as good negative samples. Thus, clicked requests are useful for training the CTR prediction model. Non-clicked requests, however, are not good training data because no ads in them are clicked or seen at all in some situations. In the case of contextual advertising, unlike in the case of sponsored

ADKDD'13 , August 11, Chicago, Illinois, U.S.A..

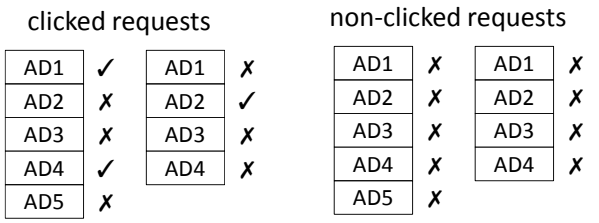

Figure 1: Examples of clicked requests and non-clicked requests. Tick denotes clicked impression and x mark represents non-clicked impression.

search on web search engine result pages, ads are placed within the content of a generic web page. Therefore, the user does not have a clear intention to see ads [4]. Results from comscore.com [11] shows that 31% of ads were not in-view, meaning they never had an opportunity to be seen. Moreover, there are many more non-clicked requests than clicked requests, so it is difficult to handle all non-clicked requests.

Figure 2 compares these two kinds of requests when changing the amounts of training data. In this comparison, we compare two methods on two kinds of testing data. Logistic regression with features described in Section 4.1.1 is used in both methods but training data differ for each method. For training data, the *trained with clicked requests* method uses only clicked requests and the *trained with both kinds of requests* method uses clicked and sampled non-clicked requests. Obviously, the *trained with clicked requests* method outperforms the other on both kinds of testing data. In addition, the improvement of the *trained with both kinds of requests* method reaches the ceiling even with increased amounts of training data. This result indicates that non-clicked requests in our datasets are not good samples.

To remedy this problem, we introduce the two-stage learning method for CTR prediction using only clicked requests. At first, because an ad requests is considered as documents related to a query in information retrieval context, a ranking model is learned by using clicked requests, and then a regression model for the CTR is built on it.

The rest of this paper is organized as follows. Section 2 provides a general overview of the contextual advertising system and learning-to-rank. Section 3 presents learning-to-rank approach for training CTR prediction model. Section 4 details the experimental setup and results. Section 5 concludes the paper by summarizing our findings and giving proposals for future work.

2. RELATED WORK

This section provides a general overview of the contextual advertising system and learning-to-rank.

2.1 Contextual Advertising System

In the contextual advertising scenario, a publisher typically reserves some space on their web page for ads, and an advertising system supplies ads that are relevant to the page content and/or user. Some works [2, 8] have taken a two-stage approach where the first stage retrieves top-K items from ad corpus with an inverted index and the second stage selects the desired top-k using brute force CTR prediction on the K retrieved items ($k \ll K$).

Because ads related to the page content are more likely rather than generic ads to provide a better user experience and thus to increase the probability of clicks, some previous works have focused on developing methods to match ads to pages. In these studies, the problem of matching ads with pages is transformed into a similarity search in a vector space. The relevance of an ad towards page content is a tf-idf score that measures the word overlap between the page content and the ad content. The works of Chakrabarti et al. [8] and Karimzadehgan et al. [18] learned weights for each word in a page and an ad using HTML tags and ad sections. Broder et al. [3] used a 6000 node semantic taxonomy in the matching function between pages and ads. In addition, Ratnaparkhi [22] introduced a page-ad probability model in which semantic relationships between page terms and ad terms are modeled with hidden classes. Murdock et al. [20] applied machine translation techniques to improve the matching between pages and ads. In the case where ad relevance cannot easily be gleaned from the page text alone, "clickable terms" approach has also been proposed [16]. This approach involves matching a website directly with a set of ad side terms, independent of the page content.

Another line of research attempts to predict the CTR of ads. These studies are not only related to contextual advertising but also sponsored search because both typically employ the pay-per-click model. Predictions of CTR for ads are generally based on a statistical model trained by using the past click data. Examples of such models are logistic regression [9, 10], probit regression [15], and boosted trees [12, 27]. The model accuracy relies greatly on the design of features. Cheng and Cantú-Paz [9] presented a framework for the personalization of click models. The authors developed user-specific and demographic-based features that reflect the click behavior of individuals and groups. The features are based on observations of search and click behaviors of a large number of users of a commercial search engine. Some recent works [1, 19, 23] proposed models to estimate conversion rates (CVR).

2.2 Learning-to-rank

Learning-to-rank has received great attention in recent years and plays a critical role in information retrieval. It aims to construct a ranking model that can sort documents for a given query from labeled training data. In a problem related to learning-to-rank, an instance is a set of objects and a label is a sorting applied over the instance. Several approaches have been proposed including the pointwise, pairwise, and listwise approaches.

Pointwise approaches transform ranking into regression or classification on the single object. In pairwise approaches, the learning problem is formalized as a binary classification problem on object pairs. Therefore, existing theories and algorithms on regression or classification can be directly applied to these approaches. However, the objective function in these approaches is formalized as minimizing errors in regression or classification rather than minimizing errors in ranking of documents. Thus, the experimental results of Burges et al. [5] show that NDCG starts to drop even if pairwise loss is still decreasing during the optimization process. To overcome this problem, listwise approaches consider object lists instead of object pairs being used as instances in learning. It is difficult to learn a ranking function to optimize the IR measures directly since they depend on the

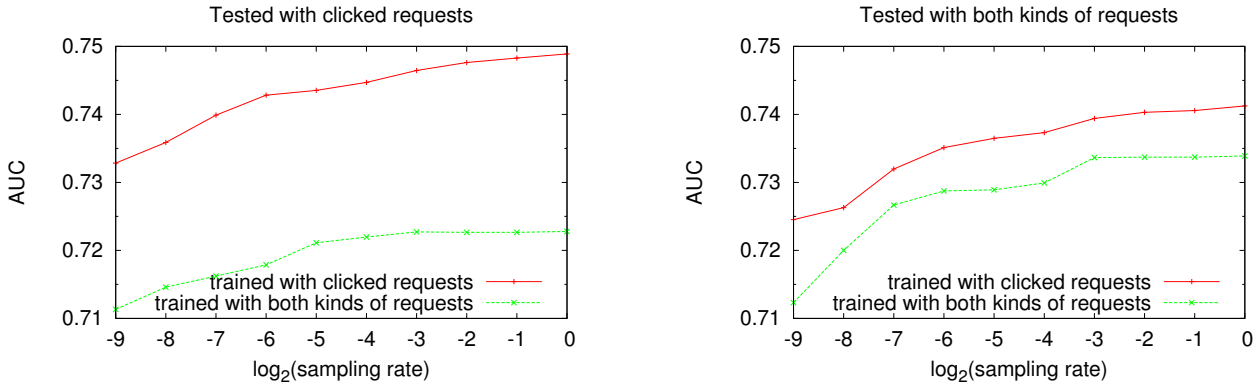

Figure 2: Comparison between clicked requests and non-clicked requests. For validation and testing data, we use only clicked requests (Left) or clicked and sampled non-clicked requests (Right).

rank and are not differentiable. To avoid the computational difficulty, Burges et al. [6] performed gradient descent on a smoothed version of the objective function. Some other listwise approaches use Genetic Programming for directly optimizing IR measures [13, 31], optimize the expectation of IR measures with the Monte Carlo sampling [28], approximate the IR measures with the functions that are easy-to-handle [26], or define loss function as an indirect way to optimize the IR evaluation metrics [7, 21].

Some works employ learning-to-rank approach for advertisements. Karimzadehgan et al. [18] proposed a method called NDCG-Annealing algorithm by integrating Simulated Annealing and downhill Simplex method to minimize the loss function associated directly to NDCG measure. The authors applied the algorithm to learn automatically the optimal weights of an ad ranking function on the basis of page-ad relevancy score. Sculley [24] proposed Combined Regression and Ranking method that optimizes regression and ranking objectives simultaneously. The author applied this combined method for predicting CTR in sponsored search advertisement. In the case of extreme class imbalance, this combination can improve regression performance due to the addition of informative ranking constraints. In this work, on the other hand, we employ the two-stage learning method for contextual advertising using only clicked requests. In behavioral targeting context, RankLDA [25] is proposed to rank audiences in accordance with their ads' click probabilities for the long tail advertisers to deliver their ads. In track 2 of KDD Cup 2012, which is a competition to predict CTR for search advertising, Wu et al. [30] built ranking models as a part of individual models to construct a final blending model. The metric for the performance of the prediction in the competition is AUC, which is concerned only about the CTR order of the testing data, so the competition participants solve a ranking problem instead of a regression problem for the real values of CTR [29]. However, in this work, we introduce a learning-to-rank approach for predicting the values of ads' CTR.

3. METHODS

In this section, we formulate two kinds of ad requests and describe a CTR prediction algorithm for contextual advertising.

3.1 Clicked Requests and Non-clicked Requests

Typically, a web publisher requests some ads simultaneously because some ads are displayed on a page at one time. In other words, one ad request r in the logs includes $N^{(r)}$ impressions of ads:

$$(\boldsymbol{x}_1^{(r)}, y_1^{(r)}), (\boldsymbol{x}_2^{(r)}, y_2^{(r)}), \ldots, (\boldsymbol{x}_{N^{(r)}}^{(r)}, y_{N^{(r)}}^{(r)})$$

Each impression of an ad consists of a pair $(\boldsymbol{x}_i^{(r)}, y_i^{(r)})$. $\boldsymbol{x}_i^{(r)}$ represents input features of the impression. The output variable $y_i^{(r)} = 1$ if a user clicked the ad, and $y_i^{(r)} = 0$ otherwise.

There are two kinds of ad requests in data R:

$$R^+ = \{r \mid \exists i (y_i^{(r)} = 1)\}$$
$$R^- = \{r \mid \forall i (y_i^{(r)} = 0)\}$$

R^+ denotes a set of ad requests that include at least one clicked impression, hence we refer to R^+ as clicked requests. R^- is called non-clicked requests since no ad requests in R^- include clicked impressions. Of course, $R = R^+ \cup R^-$ and $R^+ \cap R^- = \emptyset$.

As described in Section 1, clicked requests are useful for training, but non-clicked requests are not good samples. In addition, it is difficult to handle all non-clicked requests because non-clicked requests greatly outnumber clicked requests. To remedy these problems, we introduce a two-stage learning method for CTR prediction using only clicked requests.

3.2 Making Pairs in Clicked Requests

We describe a pairwise approach using clicked requests. Regarding ad impressions in an ad request as documents related to a query in an information retrieval context, we employ the RankSVM approach [17]. We make pairwise preferences from each clicked request r.

$$\{(\boldsymbol{x}_i^{(r)}, \boldsymbol{x}_j^{(r)}) \mid \forall i, j (y_i^{(r)} = 1 \wedge y_j^{(r)} = 0)\}$$

Figure 3 illustrates this process.

The preference $(\boldsymbol{x}_i^{(r)}, \boldsymbol{x}_j^{(r)})$ indicates that a score proportional to CTR of $\boldsymbol{x}_i^{(r)}$ is expected to be higher than that

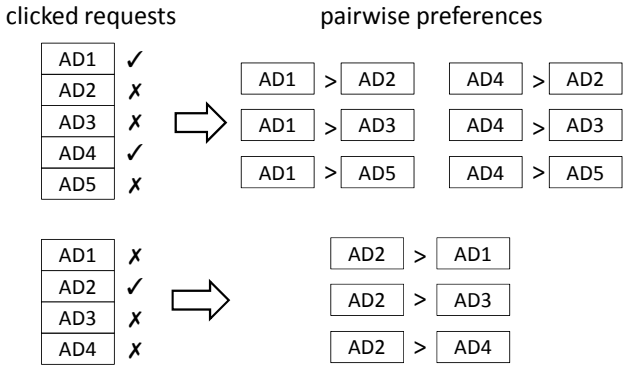

Figure 3: Making pairwise preferences from clicked requests.

of $\boldsymbol{x}_j^{(r)}$. We define the score to be a simple linear function with a weight vector $\boldsymbol{w}$ and represent the above preference as follows:

$$\begin{aligned} \text{score}(\boldsymbol{x}_i^{(r)}) > \text{score}(\boldsymbol{x}_j^{(r)}) &\Leftrightarrow \boldsymbol{w}^T\boldsymbol{x}_i^{(r)} > \boldsymbol{w}^T\boldsymbol{x}_j^{(r)} \\ &\Leftrightarrow \boldsymbol{w}^T\boldsymbol{x}_i^{(r)} - \boldsymbol{w}^T\boldsymbol{x}_j^{(r)} > 0 \\ &\Leftrightarrow \boldsymbol{w}^T(\boldsymbol{x}_i^{(r)} - \boldsymbol{x}_j^{(r)}) > 0 \end{aligned}$$

Using squared hinge loss, we define a pairwise loss function $L(\boldsymbol{w})$ as follows:

$$L(\boldsymbol{w}) = \sum_{r\in R^+} \sum_{i:y_i^{(r)}=1} \sum_{j:y_j^{(r)}=0} \max(0, 1 - \boldsymbol{w}^T(\boldsymbol{x}_i^{(r)} - \boldsymbol{x}_j^{(r)}))^2$$

Adding regularization term, we seek the weight vector $\hat{\boldsymbol{w}}$ that minimizes the following optimization problem:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C \cdot L(\boldsymbol{w}) \quad (1)$$

where $C \geq 0$ is a penalty parameter.

In our preliminary experiment by using *microMSE* described in Section 4.1.3 as an evaluation metric, we tried to use L1-loss linear SVM (hinge loss) and logistic regression (logistic loss) in addition to the above L2-loss linear SVM (squared hinge loss). In the case of using L1-loss linear SVM, training is slightly faster but prediction accuracy is no better than when L2-loss linear SVM is used. Conversely, using logistic regression achieves a little improvement but is about 20% slower. Thus, we decided to use L2-loss linear SVM in the following experiment.

3.3 Conversion to Value of CTR

Using $\boldsymbol{w}$ obtained by solving the problem (1), we can now predict the score proportional to CTR. Next, we need to convert the score into predicted CTR of an ad. Because CTR is the probability of a user clicking an ad, sigmoid function is suitable for the conversion.

$$CTR_i^{(r)} = \frac{1}{1 + \exp(-a\boldsymbol{w}^T\boldsymbol{x}_i^{(r)} + b)} \quad (2)$$

where a and b are fitting parameters. These two parameters can be obtained by solving logistic regression. In the following experiment, first we determine bias parameter b by historical CTR and then search scale parameter a to optimize target metric directly using validation data. As described in Section 4.1, historical CTR is available because it is used as an input feature.

Figure 4: Examples of YDN ads on a real estate web page.

4. EXPERIMENT

4.1 Experimental Settings

In this section, we describe the features, data sets, and models used in the experimental evaluations.

4.1.1 *Data Sets and Features*

We compare the models using the data sampled from the Yahoo! display ad network (YDN)[1] for a period of eight weeks. The data from the first four weeks are used as a training set, data from the next two weeks are used as a validation set, and data from the last two weeks are treated as a testing set. As shown in Figure 4, YDN is an ad network for the Japanese market, so all ads and pages are written in Japanese with a few exceptions.

As described in Section 3.1, each sample of the data sets is an impression of an ad and consists of pairs $(\boldsymbol{x}_i^{(r)}, y_i^{(r)})$. The output variable $y_i^{(r)} = 1$ if a user clicked the ad, and $y_i^{(r)} = 0$ otherwise. The input features $\boldsymbol{x}_i^{(r)}$ consist of the following groups, describing the aspects of an impression:

- **Web page**: tf-idf weighted terms using position in the page and HTML tags.
- **User**: categories in which the user is interested; gender and age if available, etc.
- **Ad**: tf-idf of the title and description, categories based on it, etc.
- **Web page - AD**: similarity between the web page and the ad: tf-idf in both, etc.

[1] `http://promotionalads.yahoo.co.jp/service/ydn/index.html`

Table 1: Data statistics

website	type	$\left\|R^+\right\|$	$\overline{N^{(r)}}$	$\overline{\#clicks}$
A	training	109,952	7.64	1.01
	validation	56,851	7.67	1.01
	testing	84,764	8.09	1.01
B	training	599,422	11.11	1.03
	validation	349,424	11.02	1.03
	testing	402,633	11.68	1.03
C	training	1,610,490	9.94	1.01
	validation	891,509	9.88	1.02
	testing	892,244	10.66	1.02
D	training	679,160	5.02	1.02
	validation	386,914	5.01	1.02
	testing	409,567	5.62	1.02
E	training	190,554	4.69	1.02
	validation	132,341	4.73	1.02
	testing	236,307	5.73	1.02
F	training	518,053	11.93	1.02
	validation	345,353	11.94	1.02
	testing	372,841	12.62	1.02
G	training	127,393	5.00	1.04
	validation	78,508	5.00	1.04
	testing	69,083	5.00	1.04

- **User - AD**: relevance of the ad towards the user: categories in common, etc.
- **Click statistics**: historical CTR (=clicks/impressions) of the ad and advertiser, etc.

Models we evaluate are constructed with respect to each website. The data statistics for each website are summarized in Table 1. The amount of clicked requests $\left|R^+\right|$ and average number of impressions per clicked request $\overline{N^{(r)}}$ were changed during the eight weeks. This is due to seasonal trends, changes in the budget of the advertisers, actions to achieve sales target by the publishers, etc. $\overline{\#clicks}$, which is average number of clicked impressions per clicked request, is approximately 1.

4.1.2 Baselines

We use three methods to compare with our proposed algorithm. The method introduced in Section 3 is referred to as *pairwise*.

The first method is L2-loss linear SVM with clicked requests. We refer to this method as *SVM*.

$$L_{SVM}(\boldsymbol{w}) = \sum_{r\in R^+}\sum_{i=1}^{N^{(r)}} \max(0, 1-(2y_i^{(r)}-1)\boldsymbol{w}^T\boldsymbol{x}_i^{(r)})^2$$

The second method is logistic regression with clicked requests. We call this method *LR*.

$$L_{LR}(\boldsymbol{w}) = \sum_{r\in R^+}\sum_{i=1}^{N^{(r)}} \log(1+\exp(-(2y_i^{(r)}-1)\boldsymbol{w}^T\boldsymbol{x}_i^{(r)}+w_b))$$

where w_b is a bias parameter like b in equation (2). It is also learned by training data.

The last method is *baseline*, which is logistic regression with all requests.

$$L_{baseline}(\boldsymbol{w}) = \sum_{r\in R}\sum_{i=1}^{N^{(r)}} \log(1+\exp(-(2y_i^{(r)}-1)\boldsymbol{w}^T\boldsymbol{x}_i^{(r)}+w_b))$$

For the *baseline* method, non-clicked requests are sampled approximately equal to clicked requests. In training, ad impressions in non-clicked requests are weighted by the inverse of the sampling rate.

For *pairwise* and *SVM*, sigmoid fitting is applied after obtaining the model weights. We determined parameter b in equation (2) to fit the distribution of historical CTR. After that, scale parameter a is searched for to optimize target metric using validation data directly.

Note that *pairwise* method cannot use features that have the common value in the request, such as web page and user features, because the method treats differences between two impressions in an ad request. Thus, these features disappear during the training.

Methods we described above are summarized in Table 2. In the experience, we adopt LIBLINEAR [14] to train the models.

4.1.3 Evaluation Metrics

In this paper, we consider non-clicked requests as not good samples. We thus use three evaluation metrics using only clicked requests: $microAUC$, $microMSE$, and $microLogLoss$. These metrics are defined as follows:

$$\begin{aligned} microAUC &= \frac{1}{|R^+|}\sum_{r\in R^+} AUC^{(r)}, \\ microMSE &= \frac{1}{|R^+|}\sum_{r\in R^+} MSE^{(r)}, \\ microLogLoss &= \frac{1}{|R^+|}\sum_{r\in R^+} LogLoss^{(r)}. \end{aligned}$$

$AUC^{(r)}$, $MSE^{(r)}$ and $LogLoss^{(r)}$ are calculated with respect to a request r. By using $\overline{CTR}_i^{(r)}$, which is the normalized CTR in r, $MSE^{(r)}$ and $LogLoss^{(r)}$ are defined as follows.

$$MSE^{(r)} = \frac{1}{N^{(r)}}\sum_{i=1}^{N^{(r)}}(y_i^{(r)} - \overline{CTR}_i^{(r)})^2,$$

$$\begin{aligned} &LogLoss^{(r)} \\ &= \frac{-1}{N^{(r)}}\sum_{i=1}^{N^{(r)}}\left(y_i^{(r)}\log(\overline{CTR}_i^{(r)}) + (1-y_i^{(r)})\log(1-\overline{CTR}_i^{(r)})\right), \\ &\text{where} \quad \overline{CTR}_i^{(r)} = \frac{CTR_i^{(r)}}{\sum_{j=1}^{N^{(r)}} CTR_j^{(r)}} \end{aligned}$$

Unlike $microAUC$ and conventional IR metrics such as NDCG, $microMSE$ and $microLogLoss$ are metrics which take into account not only CTR order but its value. $MSE^{(r)}$ and $LogLoss^{(r)}$ become 0 if a clicked request r includes one clicked impression whose predicted CTR is not 0 and the others are 0. As shown in Table 1, it is actually true that almost all clicked requests in the datasets include only one clicked impression.

Table 2: Method summary

method	training data	pairwise	learning method
pairwise	clicked requests	√	L2-loss linear SVM + sigmoid fitting
SVM	clicked requests	×	L2-loss linear SVM + sigmoid fitting
LR	clicked requests	×	logistic regression
baseline	clicked and sampled non-clicked requests	×	logistic regression

We normalize scores of each method by the corresponding *baseline*. All values of metrics in this paper are transformed by the equation.

$$\Delta M_{method} = \left(\frac{M_{method}}{M_{baseline}} - 1 \right) * 100$$

In our experience, similar to Trofimov et al. [27], the 0.1% difference in these metrics is small but cannot be ignored.

4.2 Experimental Results

The experimental results are summarized in Table 3. The **bold** elements indicate the best performance of the methods. We can draw several observations as follows.

By examining our three metrics, we found that *pairwise* usually outperforms other methods. Obviously, as *pairwise* directly optimizes AUC, in terms of $microAUC$, *pairwise* outperforms all other method at all sites. Also, with respect to $microMSE$ and $microLogLoss$, *pairwise* often outperforms other methods. *SVM* achieves poor results in many cases, although it optimizes target metric on the validation set in the same way as *pairwise*. Therefore, our pairwise learning-to-rank approach is better than an ordinary classification method. This is because even clicked requests are data with highly skewed class imbalance as shown in Table 1. *LR* often outperforms *baseline* and is the best in some cases. However, on site G data, *LR* is worse than *baseline* in all three metrics. On site A, while *LR* outperforms *pairwise* in terms of $microMSE$ or $microLogLoss$, *pairwise* achieves good $microAUC$ improvement. In other words, although good CTR order is obtained by pairwise approach, converting the ranking score into CTR does not work very well in these cases.

5. CONCLUSIONS

CTR prediction plays an important role in contextual advertising. It affects web publisher's revenue, advertiser's payment, and user's experience because ranking, filtering, placement, and pricing of ads are based on it.

In this paper, we introduce a two-stage CTR prediction algorithm for contextual advertising. First, a ranking model is constructed with clicked requests and then a sigmoid function converts the predicted value of the ranking model into CTR. We evaluated this approach offline on a data set based on logs from the Yahoo! display ad network. Our method is observed to achieve better results than other baselines in our three metrics.

Our future work will take the following directions. First, it is necessary to verify whether three metrics used in our experiment are appropriate. We want to carry out online evaluation in the production system and compare the results with those of offline evaluation. Furthermore, we are interested in developing a more sophisticated approach to convert the score of a ranking model into CTR of ads. Finally, we plan to investigate listwise approaches to learning CTR prediction model as well as combined regression and ranking approaches [24].

6. ACKNOWLEDGMENTS

We would like to thank our colleagues for their assistance with data collection, model evaluation, and many insightful discussions.

7. REFERENCES

[1] D. Agarwal, R. Agrawal, R. Khanna, and N. Kota. Estimating rates of rare events with multiple hierarchies through scalable log-linear models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, 2010.

[2] D. Agarwal and M. Gurevich. Fast top-k retrieval for model based recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, 2012.

[3] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, 2007.

[4] A. Broder and V. Josifovski. Introduction to computational advertising. `http://www.stanford.edu/class/msande239/`. Accessed: 15/4/2013.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, 2005.

[6] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS'06*, 2006.

[7] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, 2007.

[8] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, 2008.

[9] H. Cheng and E. Cantú-Paz. Personalized click prediction in sponsored search. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, 2010.

[10] H. Cheng, R. van Zwol, J. Azimi, E. Manavoglu, R. Zhang, Y. Zhou, and V. Navalpakkam. Multimedia features for click prediction of new ads in display

Table 3: Experimental results. All values of metrics are normalized by the corresponding *baseline*.

metrics	method	website						
		A	B	C	D	E	F	G
$\Delta microAUC$	*pairwise*	**1.9774%**	**0.7878%**	**1.2140%**	**0.8180%**	**3.8165%**	**1.0442%**	**0.3054%**
	SVM	1.1540%	-0.7677%	0.1844%	-0.4577%	-0.8980%	-0.0219%	-0.7807%
	LR	1.5192%	0.3670%	1.0027%	0.3645%	0.0779%	0.8720%	-0.4342%
$\Delta microMSE$	*pairwise*	0.0031%	**0.2970%**	**-1.3885%**	**-0.8520%**	-0.8204%	**-0.5264%**	**-0.2091%**
	SVM	1.7762%	0.8315%	-0.5191%	1.0481%	0.5681%	-0.3912%	2.5599%
	LR	**-1.1295%**	0.5745%	-1.1724%	-0.5384%	**-1.0505%**	-0.4481%	1.0575%
$\Delta microLogLoss$	*pairwise*	-0.9551%	**-0.1588%**	**-0.9594%**	**-1.3177%**	**-1.5633%**	**-0.8162%**	**-0.2064%**
	SVM	4.0535%	1.2166%	-0.5072%	1.4497%	0.5115%	-0.3520%	2.2618%
	LR	**-1.2397%**	0.4297%	-0.8698%	-0.9077%	-1.3766%	-0.7034%	0.9173%

advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, 2012.

[11] comScore Inc. comscore releases full results of vce^{TM}charter study involving 12 leading u.s. advertisers. `http://www.comscore.com/Insights/Press_Releases/2012/3/comScore_Releases_Full_Results_of_vCE_Charter_Study`. Accessed: 18/4/2013.

[12] K. S. Dave and V. Varma. Learning the click-through rate for rare/new ads from similar ads. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, 2010.

[13] H. M. de Almeida, M. A. Gonçalves, M. Cristo, and P. Calado. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, 2007.

[14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9, June 2008.

[15] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. 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*, 2010.

[16] A. Hatch, A. Bagherjeiran, and A. Ratnaparkhi. Clickable terms for contextual advertising. In *ADKDD*, 2010.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, 2002.

[18] M. Karimzadehgan, W. Li, R. Zhang, and J. Mao. A stochastic learning-to-rank algorithm and its application to contextual advertising. In *Proceedings of the 20th international conference on World wide web*, WWW '11, 2011.

[19] K.-C. Lee, B. Orten, A. Dasdan, and W. Li. Estimating conversion rate in display advertising from past performance data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, 2012.

[20] V. Murdock, M. Ciaramita, and V. Plachouras. A noisy-channel approach to contextual advertising. In *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, ADKDD '07, 2007.

[21] T. Qin, T.-Y. Liu, M.-F. Tsai, X.-D. Zhang, and H. Li. Learning to search web pages with query-level loss functions. Technical Report MSR-TR-2006-156, Microsoft Research, 2006.

[22] 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.

[23] R. Rosales, H. Cheng, and E. Manavoglu. Post-click conversion modeling and analysis for non-guaranteed delivery display advertising. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, 2012.

[24] D. Sculley. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, 2010.

[25] J. Tang, N. Liu, J. Yan, Y. Shen, S. Guo, B. Gao, S. Yan, and M. Zhang. Learning to rank audience for behavioral targeting in display ads. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, 2011.

[26] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, 2008.

[27] I. Trofimov, A. Kornetova, and V. Topinskiy. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, ADKDD '12, 2012.

[28] M. N. Volkovs and R. S. Zemel. Boltzrank: learning to maximize expected ranking gain. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 2009.

[29] X. Wang, S. Lin, D. Kong, L. Xu, Q. Yan, S. Lai, L. Wu, A. Chin, G. Zhu, H. Gao, Y. Wu, D. Bickson, Y. Du, N. Gong, C. Shu, S. Wang, K. Liu, S. Li, J. Zhao, F. Tan, and Y. Zhou. Click-through prediction for sponsored search advertising with

hybrid models. In *KDD Workshop*, 2012.

[30] K.-W. Wu, C.-S. Ferng, C.-H. Ho, A.-C. Liang, C.-H. Huang, W.-Y. Shen, J.-Y. Jiang, M.-H. Yang, T.-W. Lin, C.-P. Lee, P.-H. Kung, C.-E. Wang, T.-W. Ku, C.-Y. Ho, Y.-S. Tai, I.-K. Chen, W.-L. Huang, C.-P. Chou, T.-J. Lin, H.-J. Yang, Y.-K. Wang, C.-T. Li, S.-D. Lin, and H.-T. Lin. A two-stage ensemble of diverse models for advertisement ranking in kdd cup 2012. In *KDD Workshop*, 2012.

[31] J.-Y. Yeh, J.-Y. Lin, H.-R. Ke, and W.-P. Yang. Learning to rank for information retrieval using genetic programming, 2007.