Viewability Prediction for Online Display Ads

Chong Wang
Information Systems
New Jersey Institute of Technology
Newark, NJ 07003, USA
cw87@njit.edu

Achir Kalra
Forbes Media
499 Washington Blvd
Jersey City, NJ 07310
akalra@forbes.com

Cristian Borcea
Computer Science
New Jersey Institute of Technology
Newark, NJ 07003, USA
borcea@njit.edu

Yi Chen
Management
New Jersey Institute of Technology
Newark, NJ 07003, USA
yi.chen@njit.edu

ABSTRACT

As a massive industry, display advertising delivers advertisers' marketing messages to attract customers through graphic banners on webpages. Advertisers are charged for each view of a page that contains their ads. However, recent studies have found out that about half of the ads were actually never seen by users because they do not scroll deep enough to bring the ads in-view. Low viewability hurts financially both the advertisers and the publishers. This paper is the first to address the problem of ad viewability prediction, which can improve the performance of guaranteed ad delivery, real-time bidding, and even recommender systems. We analyze a real-life dataset from a large publisher, identify a number of features that impact the scroll depth of a given user-page pair, and propose a probabilistic latent class model that can predict the viewability of any given scroll depth for a user-page pair. The experiments demonstrate that our model outperforms comparison systems based on singular value decomposition and logistic regression. Furthermore, our model needs to be trained only once, independent of the target scroll depth, and works well in real-time.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Commercial services, Web-based services*

General Terms

Algorithms, Performance, Economics, Experimentation, Human Factors

CIKM'15 Oct 19-23, 2015, Melbourne, Australia.

Keywords

Computational Advertising; Probabilistic Prediction; Viewability; User Behaviors

1. INTRODUCTION

Online display advertising has emerged as one of the most popular forms of advertising. Studies [16] show that display advertising is generating earnings of over $63.2 billions in 2015. Online advertising involves a publisher, who integrates ads into its online content, and an advertiser, who provides ads to be displayed. Display ads can be seen in a wide range of different formats and contain items such as text, images, Flash, video, and audio. A typical display ad is shown in Figure 1: an advertiser, e.g., Audi, pays a publisher, e.g., Forbes, for space on webpages to display a banner during page views in order to attract visitors that are interested in its products. A page view happens each time a webpage is requested by a user and displayed in a browser. One display of an ad in a page view is called an ad impression, and it is considered as the basic unit of ad delivery. For instance, one view of the page in Figure 1 contains one ad impression.

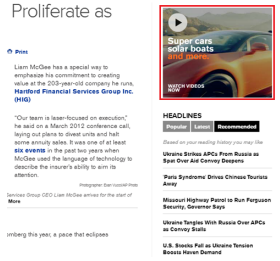


Figure 1: An Example of Display Ads

Advertisers pay for ad impressions with the expectation that their ads will be viewed, clicked on, or converted by users (e.g., the ad results in a purchase). Traditional display ad compensation is mainly based on user clicks and conversion, because they bring direct profits to the advertisers. Much research has been done for predicting click rate and conversion rate [7, 20], bid optimization [25], auctions [6], and audience selection [13].

Recently, there are growing interests by advertisers to use online display ads to raise brand awareness and to promote the visibility of the company and their products. Indeed, users like to purchase products from the brands that they trust and that they can identify. Display ads can create emotional experience that gets users excited about a brand and build trust. However, users do not typically click this type

of ads, rendering the traditional form of pricing structure based on clicks or conversion to be ineffective.

To address this, another pricing model, which pays ads by number of impressions that a publisher has served, has become popular in the display advertising market. However, a recent study [10] shows that more than half of the impressions are actually not viewed by users because they may not scroll down a page enough to view the ads. Low viewability leads to ineffective brand promotion.

In light of this, a new pricing model is emerging: pricing ads by the number of impressions that can be *viewed* by a user, instead of just being served [18]. This avoids the frustration of advertisers' concern of paying for ads that were served but not seen by users.

Not surprisingly, ads placed at different page depths have different likelihood of being viewed by a user [9]. Therefore, it is important to predict the probability that an ad at a given page depth will be shown on a user's screen, and thus be considered as *viewed*. The vertical page depth that a user scrolls to is defined as the scroll depth. Many web analytics platforms, e.g., Google Analytics, provide plugins to measure user scroll depth.

Viewability prediction is important for many applications: *Guaranteed impression delivery.* One of main ad selling methods is guaranteed delivery, in which advertisers contract publishers to buy guaranteed advertising campaigns. The contracts may fix the number of impressions, targeting criteria, price, etc. As the industry moves toward transacting on viewable impressions, advertisers may propose contracts that specify the number of guaranteed viewable impressions. Predicting ad viewability helps publishers to fulfill such contracts by placing the ads in the right impressions.

Real-time impression bidding. Advertisers can also buy impressions through real-time bidding. Given the impression context, including the user, the page, and the ad position, advertisers desire to know the probability that the ad will be in-view. Based on the viewability, advertisers can adjust the bidding price for an impression and improve ad investment effectiveness. Specifically, they can bid higher for impressions with high predicted viewability. In addition, publishers can also benefit from ad viewability prediction by adjusting the minimum prices for impressions which are offered for bidding.

Webpage layout selection. With the ad pricing standards shifting to ad viewability, viewability will become a crucial factor in page layout design, which may impact ad revenue [8]. Publishers are exploring personalized page layouts that can balance ad viewability and user experience. For example, if a user will not scroll deep, the ad slot at the bottom may be moved higher, while considering the impact on user experience.

Recommender Systems. Dwell time (i.e., the time a user spends on a page) has been regarded as an significant indicator of user interest. Recommender systems can also employ scroll depth prediction as another critical metric of user interest.

In this paper, we study the problem of predicting the probability that a user scrolls to a page depth where an ad may be placed, thus the ad can be *in-view*. To the best of our knowledge, this is the first work that tries to address viewability prediction.

Scroll depth viewability prediction is challenging. First, most users visit only several webpages on a website. It is challenging to detect user interests based on such a sparse history of user-page interaction. Second, it is hard to select the significant webpage and user features related to the user scrolling. Intuitively, page topics and user interests are regarded as influential factors. But it is non-trivial to explicitly model these features. Naturally, we may resort to latent models that utilize latent features. However, a commonly used latent model, Singular Value Decomposition (SVD), is not suitable to give probabilistic prediction on a full spectrum of scroll depths. Specifically, an SVD model can be trained with data consisting of users, pages, and whether a certain scroll depth is in-view in individual page views, and then be used to predict the viewability for that specific scroll depth. But one SVD has to be trained for each possible scroll depths. Another option is to train an SVD model with data consisting of users, pages, and the maximum page depth a user scrolls to on a page. The predicted maximum page depth can help give a binary decision for any given scroll depth (i.e., in-view or not), but it cannot give a probabilistic value for a scroll depth to be in-view, which is important to determine pricing. As a webpage typically have multiple ad slots at different page depths (and sometimes ad positions may be even dynamically determined), it may be costly to build one SVD model for every single depth.

In this paper, we first analyze a real-life dataset from a large publisher to understand user scrolling behavior. Then, in order to find out how probable a specific user is to scroll to any given page depth, we propose a viewability prediction model based on the probabilistic latent class model (PLC). Our model utilizes latent user classes and webpage classes as well as an observed scroll distribution to overcome the data sparsity issue. The output of the model is the probability that a given scroll depth is in view. Compared with a binary decision, i.e. in-view or not, a probabilistic output is very useful in optimization problems, e.g., page layout selection.

PLC has been experimentally compared with three systems: SVD, Logistic Regression (LR), and a deterministic method. The experiments show that, on average, PLC achieves a higher F1-score for both in-view and not in-view classes than the other systems. PLC has also a significantly lower prediction error than the comparison systems. Also, unlike LR and SVD, one trained PLC model can predict the viewability of any scroll depth. Thus, PLC is less costly. In addition, PLC can make predictions fast enough to be usable in real-time. Finally, PLC works well for different training datasets, with a 10-day dataset resulting in the optimal balance between training time and prediction accuracy.

In summary, the paper makes the following contributions: 1) We define the problem of viewability prediction for any scroll depth. 2) We present an empirical study of scrolling behavior using a real-life dataset. 3) We propose a novel statistical model based on PLC to predict the probability that a scroll depth will be in view. (4) We demonstrate experimentally that PLC outperforms three comparison systems.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents the results and analysis of the empirical study for user scroll depth behavior, and then describes the proposed PLC model for viewability prediction. Experimental results and insights derived from these results are presented in Section 4. The paper concludes in Section 6.

2. RELATED WORK

Researchers have investigated scrolling behavior and viewability for webpage usability evaluation. In [22, 17, 9], the authors discovered that users spend more time looking at information on the upper half of the page than the lower half, and little scrolling happens. Also, the distribution of the percentage of page content viewed by users follows a normal distribution, which is also observed in our data analysis presented in Section 3. We differ from these works in our main goal, which is viewability prediction.

Existing work [2, 12] collects scrolling behavior and uses it as an implicit indicator of user interest to measure the quality of webpage design and content. In contrast, we design an algorithm to predict the scrolling behavior for any user-webpage pair.

Several studies have attempted to predict user browsing behavior, including click [7, 21, 5, 1] and dwell time [15, 23]. For click prediction, one important application is sponsored search, i.e., ads are selected based on user queries submitted to search engines and shown along with the search results. Chen et al. [7] propose a factor model to predict if an ad shown together with search results at a specific position will be clicked on. However, this prediction is made for a given position and a query-ad pair, which does not consider the individual user as a factor. In contrast, our method makes predictions that are tailored for individual users and pages.

Wang et al. [21] learn user's click behavior from server logs in order to predict if a user will click an ad shown for the query. The authors use features extracted from the queries to represent the user search intent. In our case, search queries, which can explicitly reflect user interest, are not available. Most of the work on click prediction [5, 1] is done on the advertiser side. To predict how likely an ad is clicked, the authors collect high-dimensional features about users (e.g., private profiles), ad campaigns (e.g., ad content), and impression context. However, such data is not accessible on the publisher. Our goal is to use the publisher side data to predict page viewability.

For dwell time prediction, Liu et al. [15] fit the dwell time data with Weibull distributions and demonstrate the possibility of predicting webpage dwell time distribution from page-level features. Yi et al. [23] predict dwell time through Support Vector Regression, using the context of the webpage as features. Both methods do not consider individual user characteristics, which is an important factor of scrolling prediction.

In summary, although ad viewability and scrolling behavior have been studied, there is no existing research attempt to predict the maximum scroll depth of a user/page pair and implicitly to predict the ad viewability. In addition, existing methods for user behavior prediction cannot be easily adapted to solve the scroll depth prediction problem.

3. VIEWABILITY PREDICTION

In this section, we formally define the research problem, present the analysis of user scrolling behavior using a dataset from a large publisher which reveals several features that impact the scroll depth, and finally describe our novel viewability prediction model.

3.1 Problem Definition

Let us first introduce several important concepts used in the problem definition: 1) The *scroll depth* is the percentage of a webpage content vertically scrolled by a user. 2) The *maximum scroll depth* of a page view is how far down the page the user has scrolled during that view. The maximum scroll depth that a user u will scroll on a webpage a is denoted as x_{ua}. 3) The *target scroll depth*, denoted as X, is the page depth whose viewability an advertiser or publisher wants to predict. For instance, a publisher wants to predict the probability that an ad is in-view in a page view. In this case, the target scroll depth can be the percentage of the webpage that contains at least half of the ad. [1]

Our problem is to estimate how likely a user will be to scroll down to a target scroll depth of a webpage. Specifically, the prediction should be personalized to individual users and webpages. The proposed approach is a supervised learning technique. The inputs of the training module are historical user logs that contain the context of page views. The output is our viewability prediction model. The inputs of the prediction model are a target page depth X and a given pair of user u and webpage a, while the output is the viewability probability of X in the page view.

Problem Definition. *Given a page view, i.e., a user u and a webpage a, our goal is to predict the probability that the max scroll depth, denoted by x_{ua}, will be no less than X, i.e., $P(x_{ua} \geq X|u, a)$.*

3.2 Real-Life Dataset

We use a proprietary dataset collected over one and a half months on a large publisher's website. It contains more than 1.2 million page views and 100 thousand unique users. The dataset consists of logs of user browsing behavior captured via Javascript events. These scripts send the collected data to a server. This type of client-side approach can accurately capture users' attention even in multi-tabbed modern browsers [23].

The scroll depth is recorded according to the last row of pixels on users' screens. In this paper, we adopt 1% as the minimum unit of scroll depth; thus, the range of scroll depth is from 0% to 100%. Once a user stops scrolling and stays at a position for one second, the scroll depth is recorded in the user log. Figure 2 shows an example of the user log, in which the bottom of the user screen is at the 50% of the whole page. Thus, the scroll depth at the moment is 50%.

Figure 2: An Example of a Scroll Depth

The user log of this project includes user IDs, URLs, user agents, user geo-locations and maximum scroll depths of page views. Individual users are identified by cookies. Table 1 illustrate some of the important attributes captured in the log. Each row corresponds to a page view. For instance, the maximum scroll depth of the first page view is 72% and that of the second page view is 66%.

[1]This is in line with the definition suggested by the Interactive Advertising Bureau: a viewable display ad impression requires that a minimum of 50% of pixels be in-view for a minimum of 1 second. We do not consider the one second in-view duration.

Table 1: Example of User Log

User ID	IP	URL	Max Scroll Depth	GMT Time
001	1.3.4.5	/abc	72%	11/23/2014 11:00:00
002	7.6.9.2	/bcd	66%	11/23/2014 11:01:33

Figure 3 illustrates the distribution of max scroll depths in our user log. We observe that the distribution of the max scroll depth generally follows a normal distribution. It can also be noticed that there are very few page views whose scroll depths are less than 10%. The main reason is that the the top 10% of most webpages can be loaded on the first screen, especially on desktops. In this case, the viewability of the first 10% of webpages is almost always 1. Therefore, in this research, we mainly focus on the viewability prediction for the page depths greater than 10%.

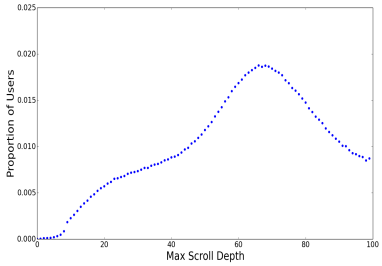


Figure 3: The Distribution of Max Scroll Depth

3.3 Features Impacting the Max Scroll Depth

We analyzed the dataset to understand which log attributes influence the scroll depth the most, with the aim of selecting these attributes as features in our prediction model.

3.3.1 *Scroll Depth vs. Device Type*

The reason that we adopted page percentage, rather than pixels, as a measure of scroll depth is because it provides a relative measure independent of device types (i.e., different devices have different screen sizes). If a user reads 50% of a page on a mobile device, while another user reads 50% of the same page on a desktop, it can be assumed that they read the same content of the page. However, this does not deny a hypothesis that devices may affect user behavior which may further influence the max scroll depth. For instance, when reading on mobile phones, users may not have enough patience and may leave the page with little scrolling.

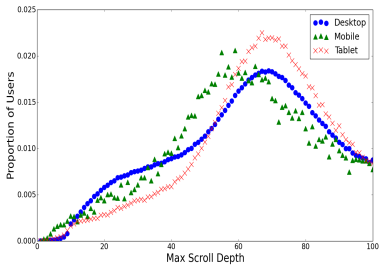


Figure 4: Distribution of Max Scroll Depth across Devices

Figure 4 illustrates the distribution of the max scroll depth across multiple devices, i.e., desktop, mobile/phone, and tablet. The device type is detected from the user agent attribute. The average max scroll depth is highest on the tablets (65.7%), followed by desktops (61.6%), and mobiles (60.2%). The possible reasons for the overall similar results across devices are: 1) The publisher's webpages are displayed in a mobile-friendly manner; 2) Flicking fingers on the screen is as easy as scrolling the wheel of a mouse [14]. Finally, we notice that mobiles, as expected, have certain page views with max scroll depth under 15%. This is very rare for desktops. The reasons for such low percentages are: 1) some pages are very long on the mobiles; 2) users close the browser tabs with loaded pages before they view these pages or stop loading the pages before they are shown, in which case the max scroll depth is zero. Although generally similar, the results exhibit a number of differences, and thus we consider the device type as a feature in our prediction model.

3.3.2 *Scroll Depth vs. Geo-location*

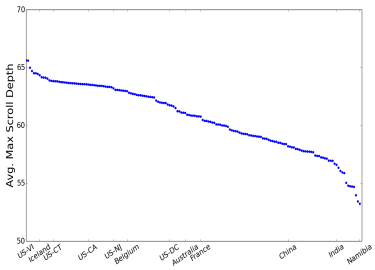


Figure 5: Average Max Scroll Depth as a Function of User Geo-location

Our user log records the countries from which the visitors connect and the US states if the visitors are from US. We filtered out the locations with sample page view sizes less than 1000. Figure 5 shows that most of the top 50 locations for max scroll depth are US states. Interestingly, visitors from U.S. Virgin Islands (65.62%) view pages the deepest, followed by New York State (65.60%) and Texas (65.49%). On the other hand, user from Namibia read the least (53.23%). In addition to user interests and reading habits, user geo-locations may also determine the connection speed, the distance from the publishers' host servers, etc. These factors, independent of users and webpages, may directly play a role on how users engage with the content. Since user geo-location is a significant factor, we consider it in our prediction model.

3.3.3 *Scroll Depth vs. Day of the Week*

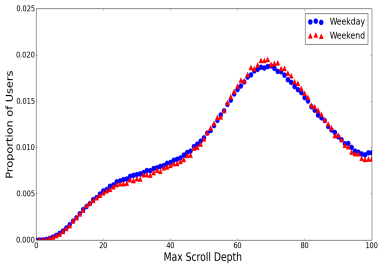


Figure 6: Distribution of Max Scroll Depth for Week Days

The day of the week and hour of the day are calculated using the local time of the user which is inferred from the user's IPs and the GMT time in the user log. Figure 6 shows that the day of the week does not have a significant impact on the scroll depth. This result contradicts past

research [24] which revealed that the day of week determines the impression volume. Thus, we do not consider the day of the week in the prediction model.

3.3.4 Scroll Depth vs. Hour of the Day

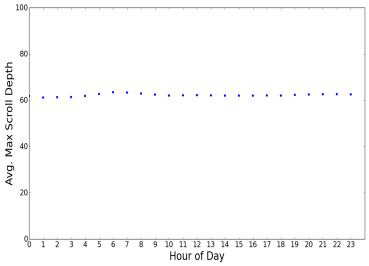


Figure 7: Distribution of Max Scroll Depth of Different Hours of the Day

One plausible hypothesis is that users may scroll deepest in the evening, after work. However, surprisingly, Figure 7 demonstrates that users seemingly perform very similar at different hours of the day. Thus, the hour of the day is not a significant factor to predict max scroll depth.

3.4 Max Scroll Depth Prediction Model

Our primary task is to infer the max scroll depth of a page view, x_{ua}, where u is the user and a is the webpage.

It is intuitive that the characteristics of individual users and webpages can be utilized to improve the performance of max scroll depth prediction models. For example, users who prefer to scroll far down on most webpages would have a higher probability to scroll down the current page. Also, features such as the ones identified in Section 3.3 (i.e., device type and geo-location) are easy to be modeled.

However, some other significant features are very hard to be captured due to lack of data and the ambiguity of user-webpage interaction. For example, pages with popular content and good design may motivate users to scroll more. But accurately modeling topic popularity and design is difficult. Other examples include user interest and psychology. Therefore, depending solely on explicit features will not lead to an accurate prediction.

In addition to feature modeling, data sparsity is another challenge. While a large publisher usually has tens of thousands of webpages, one user only visits several. Likewise, one page may be visited by a small subset of the entire user population. As a result, the user-page interaction employed in prediction could be extremely large and sparse, which brings about challenges in the prediction performance. A commonly-used solution is grouping similar users and similar webpages together and infer the prediction for a user/page pair using the known data of similar user/page pairs.

To overcome these issues, we use a latent class model [4, 3] to discover classes of users and webpages. Specifically, we build a probabilistic latent class model (PLC). The intuition behind PLC is that different latent classes of webpages and users tend to generate different levels of max scroll depths. PLC can detect classes of users and webpages that share similar patterns of max scroll depth. The class membership of each user and webpage are learned from the user log. PLC outputs the probability $P(x_{ua}|u,a)$, where $x_u a$ is the max scroll depth that a user u reaches in a page a.

In addition, PLC incorporates our finding that max scroll depths of page views follow a normal distribution, as shown in Figure 3. Particularly, given a latent user class and a webpage class, we specify the outputted max scroll depth to follow a normal distribution by modeling the conditional probability as the probability density function of the normal distribution. Formally, PLC works as follow:

$$P(x_{ua}|u,a) = \sum_{i=1}^{N_s}\sum_{j=1}^{N_p} P(s_i|u)P(p_j|a)P(x_{ua}|s_i,p_j) \quad (1)$$

where x_{ua} is the max scroll depth of a page view. N_s is the number of latent user classes, and N_p is the number of latent webpage classes. Both N_s and N_p are pre-defined as model parameters. The optimal values for these parameters can be explored by cross validation. $P(s_i|u)$ is the probability that user u belongs to the latent user class s_i, while $P(p_j|a)$ is the probability that webpage a belongs to the latent webpage class p_j. The last term, $P(x_{ua}|s_i,p_j)$, represents the probability that the max scroll depth of the page view is x_{ua}, given the latent user class s_i and webpage class p_j.

As mentioned above, the last term can be approximated by the probability density function of the normal distribution (Formula 2).

$$P(x_{ua}|s_i,p_j) = \frac{1}{\sigma_{s_ip_j}\cdot\sqrt{2\pi}} * \exp\left(-\frac{(x_{ua}-\mu_{ua})^2}{2\sigma^2_{s_ip_j}}\right) \quad (2)$$

The right side of Equation 2 is developed based on the probability density function of a normal distribution, i.e., $\frac{1}{\sigma\sqrt{2\pi}}\cdot\exp(-\frac{(x-\mu)^2}{2\sigma^2})$. The mean of the distribution, μ_{ua}, can be modeled by a regression whose features are extracted from the history of u and a as well as the context of the page view, i.e., $\mu_{ua} = \sum_m^M w_{spm} f_m^{ua}$. f_m^{ua} is the mth feature and w_{spm} is the weight of the mth feature. Each pair of latent user class s_i and latent webpage class p_j has a set of $w_{s_ip_j*}$ and $\sigma_{s_ip_j}$. M is the total number of the features.

Based on the observations presented so far, we consider seven features:

- User Features:
  1) The mean max scroll depth of all page views of u. This feature captures user browsing habits.
  2) The most recent three max scroll depths of u. This feature captures the recent scroll behavior of the user.
- Webpage Features:
  3) The mean max scroll depth of a by all users. This feature captures the overall popularity of the webpage.
  4) The most recent three max scroll depths of page views of a. This feature captures the recent scroll behavior for this webpage.
- Interaction of User and Webpage:
  5) Interaction of the mean max scroll depth of u and that of a, i.e., the product of features 1 and 3. This is a commonly-used method in statistics to capture the joint effect of two input variables, i.e., for particular user and page, the scrolling behavior depends on both.
- Page View Context:
  6) User geo-locations, which were shown to be important by our analysis of the dataset.[2]

[2]There are 172 locations in our dataset. The number of dimensions are reduced to 20 by Principal Component Analysis (PCA). We also tried feature hashing, which generates

7) Device Type (i.e., desktop, mobile, or tablet), also shown to have a certain relevance by our analysis.

These features are used to train the normal distribution of each pair of latent user class s_i and webpage class p_j. Let $\mathbf{W}$ be the collection of the weight vectors w_{sp*} of all latent user classes s and webpage classes p. $\boldsymbol{\sigma}$ is the collection of the standard deviations σ_{sp*} of all latent user classes s and webpage classes p. Specifically, the features help to iteratively determine $\mathbf{W}$ and $\boldsymbol{\sigma}$.

In Equation 1 and 2, there are several parameters ($P(s|u)$, $P(p|a)$, $\mathbf{W}$, $\boldsymbol{\sigma}$). They can be calculated by maximizing the following likelihood function:[3]

$$l\left(P(s|u), P(p|a), \mathbf{W}, \boldsymbol{\sigma}\right) = \sum_{u,a} \log\left(\sum_{i=1}^{N_s}\sum_{j=1}^{N_p} P(s_i|u)P(p_j|a)P(x_{ua}|s_i,p_j)\right) \quad (3)$$

Maximizing equation 3 determines the set of parameters. To maximize it, we adopt the Expectation Maximization (EM) algorithm, which is is widely used to solve the maximum-likelihood parameter estimation problem. The EM algorithm performs an expectation step (E-step) and a maximization step (M-step) alternatively. The E-step creates a function for the expectation of Equation 3. This function, i.e., Equation 4, is evaluated using the current estimates of the parameters. The initial values of the parameters are randomly generated.

$$P(s_i,p_j|u,a,x_{ua}) = P(s_i|u)P(p_j|a)\cdot\frac{1}{\sigma_{s_ip_j}\cdot\sqrt{2\pi}}\cdot \exp\left(-\frac{(x_{ua}-\sum_m^M w_{s_ip_jm}f_m^{ua})^2}{2\sigma_{s_ip_j}^2}\right) \quad (4)$$

The M-step updates the parameters in Equation 4, which can maximize Equation 3. The M-step updates the value of each parameter based on the result of the E-step of each iteration. The updated ${w_{s_ip_j}}^*$ of each iteration in Equation 7 can be determined by Limited-memory BFGS (L-BFGS), an optimization algorithm in the family of quasi-Newton methods. Equation 8 is the closed form of standard deviation of normal distributions.

$$P(s_i|u)^* \propto \sum_{p,a} P(s_i,p|u,a,x_{ua}) \quad (5)$$

$$P(p_j|a)^* \propto \sum_{s,u} P(s_j,p|u,a,x_{ua}) \quad (6)$$

$$w^*_{s_ip_j*} \propto \underset{w_{s_ip_j}}{argmax}\{-\sum_{u,a} P(s_i|u)P(p_j|a)\cdot [\frac{(x_{ua}-\sum_m^M w_{s_ip_jm}f_m^{ua})^2}{2\sigma_{s_ip_j}^2} + \log\sigma_{s_ip_j} + \log\sqrt{2\pi}]\} \quad (7)$$

similar performance.

[3]Since Equation 2 is an exponential function, we do a log transformation in the likelihood function in order to approximate the parameters. Note that there is no loss of information in using a log transformation because the log is a one-to-one function.

$$\sigma^*_{s_ip_j} \propto \sqrt{\frac{\sum_{ua} P(s_i|u)P(p_j|a)(x_{ua}-\sum_m^M w_{s_jp_jm}f_m^{ua})^2}{\sum_{ua} P(s_i|u)P(p_j|a)}} \quad (8)$$

The EM iterations stop if the max ratio defined below is not greater than a pre-defined threshold, which is set to 10^{-3} in our experiments. For each m in M,

$$\left|w_{s_ip_jm,t} - w_{s_ip_jm,t-1}\right| / w_{s_ip_jm,t} < 10^{-3} \quad (9)$$

After convergence, the PLC model with the optimal parameters can predict $P(x_{ua}|u,a)$, i.e., the probability of any target max scroll depth x_{ua} of a user/webpage pair. Section 3.5 uses this probability to predict the viewability of any target scroll depth. Similarly, this model can be applied to recommender systems, as mentioned in Section 1. The predicted max scroll depth x_{ua} reflects the interest of the user u in the webpage a.

3.5 Viewability Prediction for a Target Scroll Depth

Given a target scroll depth X and a user/webpage pair, the trained PLC model computes the probability that the max scroll depth will be X, i.e., $P(x_{ua} = X|u,a)$. As stated in the problem definition, the goal of this project is to predict the probability that a given scroll depth will be in view, i.e., $P(x_{ua} \geq X|u,a)$. Therefore, we integrate $P(x_{ua}|u,a)$ from X to 100%, as shown in Equation 10. The result is the probability that the max scroll depth of the page view will be greater or equal to the target scroll depth X. This means the max scroll depth x_{ua} is at a page percentage no less than X. The upper bound of the max scroll depth is 100%, i.e., the bottom of a page.

$$P(x_{ua} \geq X|u,a) = \int_X^{100\%} P(x_{ua}|u,a)dx_{ua} \quad (10)$$

4. EXPERIMENTAL EVALUATION

4.1 Experiment Datasets

To evaluate the proposed method, we use Forbes' user browsing log as described in Section 3. The user log is split into three sets of training and testing data, as shown in Table 2. This was done to avoid bias. The experimental results are reported by taking the average over the the three sets. On average, there are 31K+ unique users who generated 300K+ page views in a 10 days training set and 23K+ page views in a 2 days testing set. We run the experiments on a computer with Intel Core i7 3.6GHz and 32GB of memory.

Table 2: Training and Test Data Partitioning

Set#	Training Data (10d)	Testing Data (2d)
1	11/01/2014-11/10/2014	11/11/2014-11/12/2014
2	11/13/2014-11/22/2014	11/23/2014-11/24/2014
3	11/25/2014-12/4/2014	12/5/2014-12/6/2014

4.2 Comparison Systems

We compare the performance of the proposed model (PLC) with three other system described below: a deterministic method, a logistic regression (LR) system, and a singular value decomposition (SVD) system. implement several comparison systems. PLC and LR are implemented in Java, while SVD is built in C++. We use MongoDB to store pre-processed user logs.

Deterministic Method (DET): We compute the proportion of the page views whose max scroll depths are greater or equal with the target scroll depth X in each training set. This proportion is the prediction for all page views given X. For instance, $P(x_{ua} \geq 30\%|u, a)$ is 0.8953 means that the viewability x_{ua} for all test page views is 0.8953. Formally:

$$P(x_{ua} \geq X|u, a) = \frac{\#pageviews\ whose\ x_{ua} \geq X}{\#pageviews}$$

Logistic Regression (LR): We build an LR model based on the Stanford NLP API. Since one LR model cannot predict for every given target scroll depth, we train an LR model for each target scroll depth. We use the same set of input features as those used to train PLC. The target variable is 1 or 0, i.e., if a page scroll x_{ua} is not less than X, then target variable is 1; otherwise it is 0. When testing, given the features vector of a test page view, the LR model outputs the probability that X is in-view, i.e., $P(x_{ua} \geq X|u, a)$. This probability can be further converted into a binary decision.

Singular Value Decomposition (SVD): In addition to dimension reduction, SVD is often used to predict a target variable based on historical data. For any $M * N$ matrix A of rank r, SVD can decompose it as $A = U \sum V^T$. U is a $M * M$ orthogonal matrix that spans the "column space". V is a $N * N$ orthogonal matrix that spans the "row space". $\sum$ is a $M * N$ diagonal matrix whose first r entries are the nonzero singular values of A. Using matrix factorization, SVD maps both row items (e.g., users) and column items (e.g., pages) to a joint latent factor space, such that the interactions of row items and column items are modeled as inner products in that space. In our case, it generates a vector to represent each user or page. The dot product of a user vector and a webpage vector is the prediction of their interaction. Unlike PLC, SVD does not utilize the distribution of max scroll depth and the explicit features of page views.

Our SVD model implementation is based on libFM [19]. The number of factors is set to 8, as suggested in the manual. The matrix A is a user-webpage matrix. Each cell value is either 1 or 0, i.e., whether X is in-view or not. The output for a page view is a value between 0 and 1, which is treated as the probability that X is in-view. This probability can be converted into a binary decision. Similar to LR, we build an SVD model for each X.

4.3 Metrics

The main metrics we adopt are the Root-Mean-Square Deviation (RMSD) and the F1-score of class 0 (i.e., given scroll depth not in-view) and class 1 (i.e., given scroll depth in-view). We also compare the methods using the precision and recall metrics.

RMSD: The RMSD measures the differences between the values predicted by a model, $\hat{y}_i$, and the values actually observed, y_i. It is widely used in various research fields and is defined as the square root of the mean square error:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

where N is the number of test page views. y_i is the ground truth of the ith page view. If the target scroll depth X is in-view, $y_i = 1$; otherwise, $y_i = 0$. $\hat{y}_i$ is the probabilistic prediction of the ith page view, i.e., $\hat{y}_i \in [0, 1]$. RMSD serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of the predictive power of a method. Thus, the lower RMSD is, the better the prediction performance.

Precision, Recall and F1-score: The probability that X is in-view can be converted to 0 or 1, i.e., if it is greater or equal than 0.5, then X is in-view; otherwise, X is not in-view. Thus, the probabilistic prediction problem can be considered a binary classification problem as well. Hence, precision, recall, and F1-score can be used to compare the models. The precision of a class is the number of page views correctly labelled as belonging to the class divided by the total number of page views labelled as belonging to the class. High precision means high true positive rate and low false positive rate. The recall of a class is the number of page views correctly labelled as belonging to the class divided by the total number of page views that belong to the class. High recall means high true positive rate and low false negative rate. The F1-score of a class is the harmonic mean of the precision and recall of the corresponding class.

4.4 Effect of Parameter Combination

We investigate the performance of PLC with different combinations of the two parameters, N_s and N_p, shown in Equation 1. N_s is the number of latent user classes, while N_p is the number of latent webpage classes. Since there is an ad slot located at the 60% page depth on the real webpages analyzed, we take 60% as the target scroll depth X. We adopt grid search and random search to find the optimal parameters. For grid search, we try all combinations of $N_s \in [2, 12]$ and $N_p \in [2, 12]$. For random search, we try 20 combinations of $N_s \in [2, 30]$ and $N_s \in [2, 30]$ which are not included in the grid search. The range of obtained RMSDs is [0.3637, 0.3683].

Table 3: RMSDs of Different Parameter Pairs

RMSD	N_p=4	N_p=5	N_p=6	N_p=7	N_p=8	N_p=9
N_s=4	0.3681	0.3672	0.3678	0.3678	0.3676	0.3659
N_s=5	0.3671	**0.3691**	0.3678	0.3686	0.3675	0.3663
N_s=6	0.3679	0.3676	0.3678	0.3679	0.3671	0.3659
N_s=7	0.3674	0.3679	0.3672	0.3672	0.3645	0.3656
N_s=8	0.3675	0.3678	0.3663	0.3640	0.3672	0.3660
N_s=9	0.3678	0.3671	0.3652	0.3652	0.3638	0.3663
N_s=10	0.3671	0.3673	0.3649	0.3639	0.3644	0.3646
N_s=11	0.3657	0.3644	0.3637	**0.3631**	0.3638	0.3643
N_s=12	0.3640	0.3637	0.3634	0.3636	0.3645	0.3644

Table 3 shows the 5-fold cross validation RMSD results for different N_s and N_p combinations. For the sake of brevity, we only present partial results which contain the best and the worst performance. We observe that different combinations do not largely influence the performance, with the difference between the best and the worst results being only 0.006. Most parameter combinations generate similar values for precision, recall, and F1-score, respectively.

4.5 RMSD Comparison

The goal of this experiment is to test the performance of the models with different target scroll depths. Since generally the top 10% of a page can be shown in the first screen without the user performing any scrolling, we set the range of the target scroll depth to the interval [0.1, 1].

Figure 8 plots the RMSD comparison for the four systems. The results show that PLC significantly outperforms the three comparison systems. The RMSD performance of PLC at all Xs is averagely 10%, and 17% at maximum,

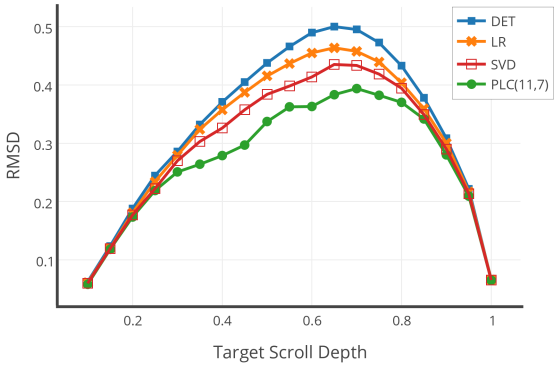


Figure 8: RMSD Performance

better than the second best system, SVD. All models have better performance near the top and bottom of a webpage than in the middle. The reasons for the top of the pages is that most pages are in-view at scroll depths such as [0.1, 0.2]. Being trained by such skewed data, most probabilistic outputs of the models are closer to 0 than 1. Although they may commit mistakes on the cases that are not in-view, the average RMSDs are still relatively low.

The prediction becomes harder with X moving toward the middle of the pages. Intuitively, the models are more prone to making incorrect predictions. Thus, RMSDs in this interval are higher than those in the two tails. Nevertheless, PLC performs substantially better than the other systems within this challenging interval. Due to the difficulty of capturing all the significant features, logistic regression does not perform as well as SVD and PLC, which identify latent features or latent classes, respectively.

Although RMSD reflects the deviation between probabilistic prediction and ground truth, it cannot tell the whole story of the performance. For example, let us assume there are 100 page views. Given a certain X, the ground truth tells that 99 belong to the not-in-view class and one belongs to the in-view class. A naive model makes the same prediction, which is 0, all the time. Thus, RMSD for this naive model at X is 0.1, which looks decent. However, such a good RMSD hides the inability of the model to recognize in-view instances. To overcome this issue, we adopt precision, recall, and F1-score to further evaluate our model.

4.6 Precision, Recall, and F1-score Comparison

Avoiding both false positives and false negatives can improve investment effectiveness for advertisers and increase the ad revenue for publishers. Therefore, identifying both in-view and not in-view impressions is equally important. Two practical examples illustrate this goal: (1) since the viewability of the page bottoms tends to be low, it is important to recognize when the page bottoms are actually in-view; (2) relatively high viewability of the page tops leads to expectations that ads at top are always in-view; however, this is not always the case, and it is very helpful to identify those pages whose tops are not in-view.

Figure 9 shows the precision, recall, and F1 score of both class 0 and 1 (i.e., not in-view and in-view). Overall, PLC performs the best among the four systems. The performance for class 1 is high when X is set in the interval [0.1, 0.6] because the top of most pages are in-view. Although it is more challenging to recognize the page views whose top is not in-view, PLC classifies these page views the best because its precision and recall for class 0 in the interval [0.1, 0.6] are the highest. Likewise, although it is difficult to detect the page views whose bottoms are in-view, PLC has the highest precision and recall for class 1 within [0.6, 1].

PLC has relatively low recall for class 1 in the interval [0.3 0.6] because it tends to boldly classify more page views to class 0 than the other systems. Most of these predictions are correct, (i.e., true negatives), while just a few are wrong (i.e., false negatives). The correct predictions increase the precision and recall for class 0, but the wrong predictions inevitably decrease the recall for class 1 since fewer page views are classified into class 1. This also explains why PLC's precision for class 1 is the highest in the interval [0.3, 0.6]. In the interval [0.6, 1], these observations are even more apparent. At the cost of sacrificing the recall for class 0, PLC achieves decent performance on the precision for both classes as well as the recall for class 1.

The differences among the models in Figure 9 are not as substantial as those in Figure 8 because RMSD is a more sensitive metric. For instance, given a page view whose X is in-view according to the ground truth, the probabilistic prediction of PLC is 0.8, while that of LR is 0.6. Both methods have the same evaluation results on the classification metrics because the outputs are greater than 0.5. But their performance can be distinguished when looking at RMSD: PLC's RMSD is 0.2, while LR's is 0.4.

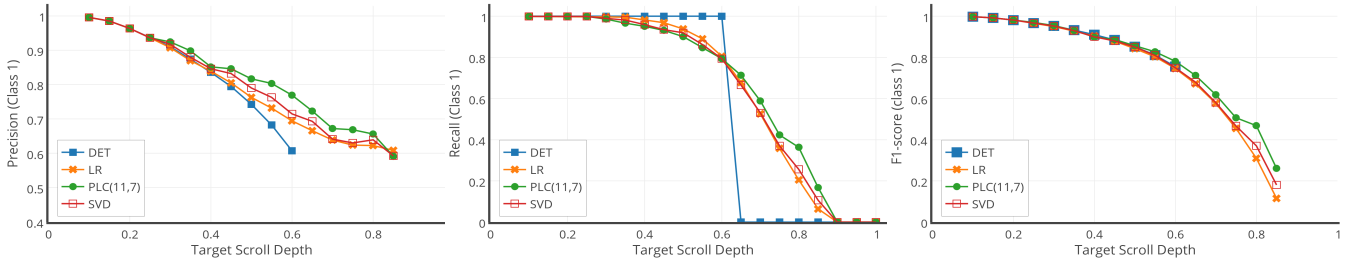
LR, SVD, and PLC do not have precision results for class 1 in the interval [0.9, 1] because no page view is classified into class 1. Thus, a precision value cannot be calculated because the number of page views labeled in class 1 acts as the denominator in the precision formula and is 0 in this case. For the same reason, the recall for class 1 is 0 in this interval and no F1-score for class 1 can be computed for this interval. A similar behavior happens for class 0 in the interval [0.1, 0.2].

The reason that no page view is classified into class 1 within [0.9, 1] is that the distributions of the two classes are very skewed in the interval. Particularly, a large majority of page views are not in-view. Such imbalanced data precludes statistical methods like ours to work appropriately [11]. Essentially, the classifiers cannot learn well from the skewed data because the training examples are scarce. To overcome this issue, we have tried simple under/over-sampling. But inevitably, the precision has largely decreased. Therefore, mitigating data imbalance remains a task for future work.

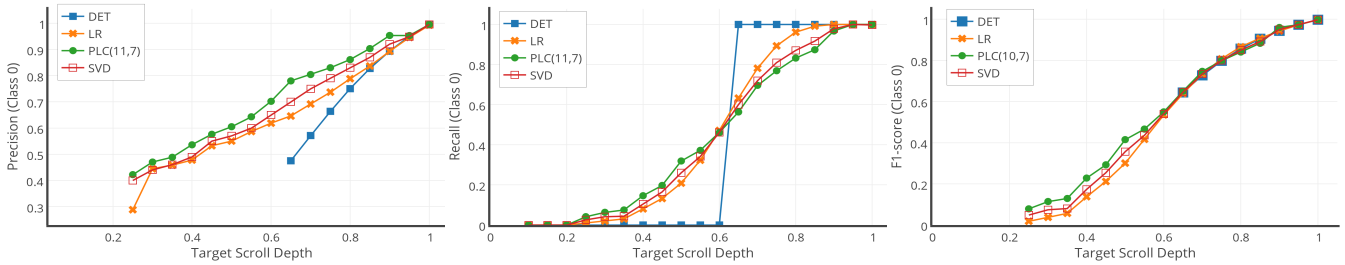
Note that DET is not impacted by imbalanced data because it always makes the same decision for all test page views given an X. It works as well as the other methods in the interval [0.1, 0.2] and [0.9, 1]. Since DET is much simpler and faster, a practical suggestion on viewability prediction is to use DET to predict the viewability of scroll depths in [0.1, 0.2] and [0.9, 1] intervals, while PLC should be employed to predict in [0.2, 0.8] interval.

4.7 Runtime Comparison

Figure 10 shows the runtime comparison for PLC, LR, and SVD. In this experiment, we build one PLC model to predict the viewability of all target scroll depths from 10% to 100%. (step = 5%, so 19 scroll depths). However, for LR and SVD, we build 19 models (the step is 5% for the interval 10% to 100%). Therefore, the time for PLC includes one training, while the time for LR and SVD is the sum of 19 trainings. We do not include DET because it does not involve training

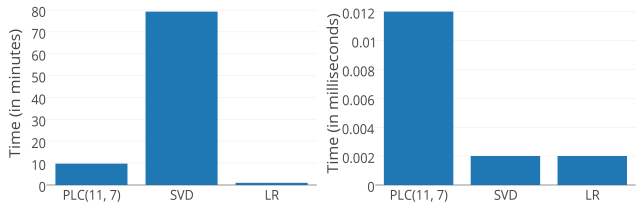


(a) Precision of Class 1 (b) Recall of Class 1 (c) F1 Score of Class 1



(d) Precision of Class 0 (e) Recall of Class 0 (f) F1 Score of Class 0

Figure 9: Classification Performance Comparison



(a) Avg. Training Time for Entire Training Set **(b) Avg. Testing Time for One Testing Page View**

Figure 10: Runtime Comparison

and makes consistent predictions for all page views for a given X (i.e.. its training and testing runtime are almost 0).

The results show that the training time of LR is much lower than those of PLC and SVD because LR does not have to learn any latent patterns from data. Intuitively, learning and applying more latent user classes and webpage classes takes more time. Since PLC performs better in terms of prediction accuracy, its training time is reasonable, especially compared to SVD. Let us also note that training can be done offline.

The results also show that PLC needs more time to make a prediction. However, the absolute value is very low (i.e., 0.012 ms). As an exchange-sold ad is often sold in 200 milliseconds, the PLC prediction time can easily be afforded for real-time predictions of incoming pages.

4.8 PLC Performance on Different Training Data Sizes

Table 4: Dataset Partitions with Different Sizes

Training Data	Testing Data (2d)
11/10/2014 (1d)	11/11/2014-11/12/2014
11/01/2014-11/10/2014 (10d)	
10/22/2014-11/10/2014 (20d)	
10/12/2014-11/10/2014 (30d)	

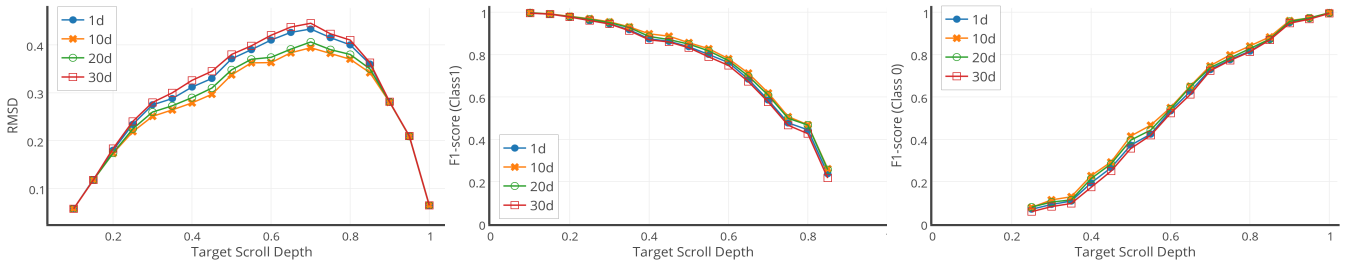
To test the impact of different training data sizes on PLC's performance, we re-partition the dataset by fixing the testing dates and varying the training data sizes, as shown in Table 4. All models share the common parameter pair, $N_s = 11$ and $N_p = 7$ According to Figure 11, the PLC results are almost the same for F1 scores. However, the results are distinguishable for RMSD, as this is a more sensitive metric. RMSD for PLC(30d) is slightly worse than the others. A possible reason is that the user interest may change over time a longer period of time and subsequently hurt the prediction performance. The performance of PLC(1d) is not as good as those of PLC(10d) and PLC(20d) because it utilizes much less user and webpage history. Generally, PLC(10d) and PLC(20d) have very similar performance. The former should be preferred in practice because less data are required for training.

5. ACKNOWLEDGEMENT

This work is partially supported by National Science Foundation (NSF) CAREER Award IIS-0845647, NSF Grant No. CNS 1409523, Google Cloud Service Award and the Leir Charitable Foundations. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

6. CONCLUSIONS

To the best of our knowledge, this paper is the first to study the problem of predicting the viewability probability for a given scroll depth and a user/webpage pair. Solving this issue is of great value to online advertisers and publishers because it will allow them to invest more effectively in advertising and increase their revenue, respectively. We presented PLC, a probabilistic latent class model, that is trained only once to predict the viewability for any given scroll depth. The model includes a number of features identified from our analysis of a dataset from a large online publisher to have an impact on the maximum scroll depth, such as user geo-location and device type. The experimental re-



(a) RMSD **(b) F1 Score for Class 1** **(c) F1 Score for Class 0**

Figure 11: Performance Comparison of Different Training Data Sizes

sults show that PLC has higher prediction accuracy than three system used for comparison. The results also demonstrate that PLC can be used in real-time and works well for different training datasets.

In the future work, we plan to work on collecting data about webpage content, investigating its effect on viewability, and incorporating that information in our model for viewability prediction. We will continue improving the proposed model to handle unbalanced data.

7. REFERENCES

[1] D. Agarwal, B. Long, J. Traupman, D. Xin, and L. Zhang. Laser: a scalable response prediction platform for online advertising. In *ACM WSDM'14*, pages 173–182, 2014.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *ACM SIGIR'06*, pages 19–26, 2006.

[3] S. Cetintas, D. Chen, and L. Si. Forecasting user visits for online display advertising. *Information retrieval*, 16(3):369–390, 2013.

[4] S. Cetintas, D. Chen, L. Si, B. Shen, and Z. Datbayev. Forecasting counts of user visits for online display advertising with probabilistic latent class models. In *ACM SIGIR'11*, pages 1217–1218, 2011.

[5] O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):61, 2014.

[6] W. Chen, D. He, T.-Y. Liu, T. Qin, Y. Tao, and L. Wang. Generalized second price auction with probabilistic broad match. In *ACM EC'14*, pages 39–56, 2014.

[7] Y. Chen and T. W. Yan. Position-normalized click prediction in search advertising. In *Proceedings of the 18th ACM SIGKDD*, pages 795–803, 2012.

[8] H. Cheng, E. Manavoglu, Y. Cui, R. Zhang, and J. Mao. Dynamic ad layout revenue optimization for display advertising. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, page 9, 2012.

[9] S. Flosi, G. Fulgoni, and A. Vollman. if an advertisement runs online and no one sees it, is it still an ad? *Journal of Advertising Research*, 2013.

[10] Google. The importance of being seen. http://think.storage.googleapis.com/docs/the-importance-of-being-seen_study.pdf.

[11] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[12] M. Holub and M. Bielikova. Estimation of user interest in visited web page. In *WWW'10*, pages 1111–1112, 2010.

[13] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, L. Garcia-Pueyo, and J. Yuan. Focused matrix factorization for audience selection in display advertising. In *IEEE ICDE'13*, pages 386–397, 2013.

[14] S. Kyle. Experimenting in loyalty conversion with wnyc: Achieving mobile-desktop parity. http://blog.chartbeat.com/2013/10/07/experimenting-loyalty-conversion-wnyc-achieving-mobile-desktop-parity/.

[15] C. Liu, R. W. White, and S. Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *ACM SIGIR'10*, pages 379–386, 2010.

[16] I. Lunden. Internet ad spend to reach $121b in 2014. http://techcrunch.com/2014/04/07/internet-ad-spend-to-reach-121b-in-2014-23-of-537b-total-ad-spend-ad-tech-gives-display-a-boost-over-search/.

[17] F. Manjoo. You won't finish this article. *Slate*, 2013.

[18] M. Mareck. Is online audience measurement coming of age? *Research World*, 2015(51):16–19, 2015.

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

[20] R. Rosales, H. Cheng, and E. Manavoglu. Post-click conversion modeling and analysis for non-guaranteed delivery display advertising. In *ACM WSDM'12*, pages 293–302, 2012.

[21] C.-J. Wang and H.-H. Chen. Learning user behaviors for advertisements click prediction. In *ACM SIGIR'11 Workshop on Internet Advertising*, pages 1–6, 2011.

[22] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Not quite the average: An empirical study of web use. *ACM Transactions on the Web (TWEB)*, 2(1):5, 2008.

[23] X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan. Beyond clicks: dwell time for personalization. In *ACM Recsys'15*, pages 113–120, 2014.

[24] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 3, 2013.

[25] W. Zhang, S. Yuan, and J. Wang. Optimal real-time bidding for display advertising. In *ACM SIGKDD'14*, pages 1077–1086, 2014.