International Journal of Pure and Applied Mathematics
Volume 119 No. 14 2018, 1433-1443
ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

ADVERTISEMENT VIEWABILITY PREDICTION ON THE WEB

Malarvizhi R[1], Anbuthiruvarangan.K[2] , Srinivasan.A[2]
[1]PG Student, [2]Assistant Professor, Department of CSE, Thangavelu Engineering College, Chennai.
[2]Department of CSE,Vel Tech Rangarajan Dr.Saguunthala R&D Institute of Science and Technology Chennai

ABSTRACT

Advertisement act as a vital role in every industry and it helps in growth of the industry. The advertisement that are published are not seen properly by the user just because of insufficient clicking. So we have proposed, two probabilistic latent class models(PLC).This helps view ability of the user that he has scrolled it is called as scroll depth of the user page pair it is based on users own interest. So they can look at the advertisement and buy the product. So it improves the credit ability of the online advertisers. So it leads to more investment on advertising by the growing and upcoming industries to earn more money.

Keywords: Computationale Advertising,Viewability Prediction, User Behaviour.

INTRODUCTION

Online display advertising has arose as one of the most famous forms of advertising. Studies [1] show that publishing advertising has generated earnings of over $63.2 billion in 2015. Internet advertising involves a publisher, who integrates advertisements into its online content, and an advertiser, who provides advertisements to be published. Display ads can be seen in a wide range of various formats and contain items such as text, images, Flash, video, and audio. In display advertising, an advertiser wages a publisher for space on webpages to display a banner during page views in order to impress visitors who are interested in its products. A page view happens each time a webpage is requested by a customer and displayed in internet. One display of an advertisement in a page view is called an ad impression, and it is considered the basic unit of advertisement delivery.

Advertiser's wages for ad impressions with the expectation that their advertisements will be viewed, clicked on, or converted by customers (e.g., the ad results in a purchase). Old fashioned display advertisement compensation is mainly based on customer clicks and changes, because they bring direct income to the advertisers. Much research has been done for probabilistic analysis on click rate and conversion rate [2], bid optimization [3], and public sales of

companies and their products. Certainly, customers like to purchase products from the varieties that they recognize and trust.

Display advertisements can make an expressive experience that gets customers surprised about varieties and creates a trust. However, customers do not naturally clicks this type of advertisements, version the old fashioned form of pricing structure based on clicks or conversion to be inefficient.

To note this problem, another pricing model, which wages advertisements by the number of rendering imitations that a publisher has served, has become popular in the display advertising field. However, a modern study [5] shows that more than half number of the imitations are actually not seen by customers because they do not scroll down a page insufficient to view the advertisements. Low view ability leads to inefficient brand and Advertising.

Therefore, a new pricing model is growing: pricing advertisement by the number of imitations that can be seen by a customer, instead of just being used [6]. This eliminates the stress of advertisers, who concern about giving for advertisement that were served but not viewed by customers.
Not amazingly, advertisements located at different page depths have different probabilities of being seen by a customer [7]. Therefore, it is mandatory to judge the probability that an advertisement at a known page depth will be viewed on a customer's screen, and thus be measured as seen. The vertical page depth that a customer scrolls to is definite as the scroll depth.

1. REVIEW OF RELATED WORK

Researchers have investigated scrolling behavior and view ability for webpage usability evaluation. In [7], [9], [10], the authors discover that users spend more time looking at information on the upper half of the page than the lower half. Also, the distribution of the percentage of content viewed by users follows a Gaussian-like distribution. We differ from these works in our main goal: view ability prediction. Existing work [11], [12] collects scrolling behavior and uses it as an implicit indicator of user interests to measure webpage quality. In contrast, we design algorithms to predict the scrolling behavior for any user-webpage pair.

Several studies have attempted to predict user browsing behavior, including click [2], [13], [14], [15], [16] and dwell time [17], [18]. The existing methods on click prediction are not applicable in our application. They rely heavily on side information (e.g., user profile, and users' queries and tweets) [16] in order to detect what the user is looking for and thereby suggest the items that are more likely to be clicked on. In our application, on the other hand, there is no such kind of explicit indicators of user information needs and detailed user profile. Wang et al. [13] 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 Internet. In our case, search queries, which can explicitly reflect user interests, are not available.

Most of the work on click prediction [14], [15] is done on the advertiser side, based on high-dimensional features about users (e.g., private profiles), ad campaigns (e.g., ad content), and impression context. However, such data is not accessible at the publisher side. Our goal is to use the publisher data to predict page depth view ability. In addition, Chen et al. [2] 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, but does not consider the individual users as a factor. In contrast, our methods make predictions that are tailored for individual users and pages. Furthermore, compared with other user responses, scrolling is a more casual behavior because users may terminate the viewing process at any time [19]. In contrast, users do not easily click an item. In other words, clicking is more deliberate, while scrolling is more casual.

For dwell time prediction, Liu et al. [17] 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. [18] predict dwell time through Support Vector Regression, using the context of the webpage as features. However, both methods do not consider individual user characteristics, an important factor of scrolling prediction.

Our models are also related to meta-level hybrid recommender systems [20], which typically cascade a content based and a collaborative system. However, existing studies [21], [22], [23] in meta-level category are not applicable in our case. They either require pre-knowledge about user detailed profiles, e.g., gender and age, or pairwise preference of items, which are not available in our case.

In summary, there is no existing research attempt to predict the maximum scroll depth of a user/page pair and to predict ad view ability. In addition, existing methods for user behavior prediction cannot be easily adapted to solve the scrolling behavior prediction problem. Exploring such a problem, our pilot work [24] analyzes a real-life dataset, identifies the features that impacts scrolling behavior, and proposes a probabilistic latent class model that predicts the view ability of any given scroll depth for a page view using constant user and webpage memberships 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 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 view ability 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 view ability probability of X in the page view.

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

content and good design may motivate users to scroll more. But accurately modeling topic popularity and webpage design is difficult. Other examples include user interests and psychology. Therefore, depending solely on explicit features will not lead to 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 sparse, which brings about challenges in the prediction performance. A widely-used solution is grouping similar users and similar webpages together and inferring the prediction for a user-page pair using the known data of similar user-page pairs.

2. MATERIALS AND METHODS

Earlier mechanisms did not work because of lack in query terms. So, we have proposed the process, **Real-time impression bidding:** Investors can also get imitations through real-time bidding. Given the imitation context, including the customer, the page, and the advertisement placement, advertisers desire to know the probability that the advertisement will be in-view. Based on aptitude, advertisers can adjust the bidding price for impression and improve advertisement investment effectiveness. Specifically, they can bid higher for imitations with high predicted view ability. In addition, publishers can also benefit from advertisement view ability prediction by adjusting the least prices for imitations which are rendered for bidding.

Webpage layout selection: Aptitude is expected to become a critical factor in page layout design, which may react advertisement revenue. Investors are finding personalized page layouts that can balance ad view ability and customer's experience. For example, if a customer will not scroll deep, the advertisement duration at the bottom of the page may be moved higher, while considering the reaction on customer experience.

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

2.1 Proposed System

Web pages can be used to improve the skillset of max scroll depth prediction models. The customers who prefer to scroll last down on most browser would have a higher probability to scroll down the current page. In this project we are implementing two proposed PLC models to implement the essentially better than the other models within this interesting interval. By calculating offline the credentials of users and browser belonging to latent user and webpage classes, PLC const predicts the view ability of any target scroll depth in a page view. However, customer and webpage memberships in reality can be dynamic during the internet process, since customer needs and page popularity keep changing. The advantage of prescribed system, Both PLC models have essentially better probabilistic performance than the comparative systems. The PLC with dynamic memberships can better adapt to the shift of customer needs and browser attractiveness and has less memory consumption. Predict not only the depth of the scroll page and also the ip address, URL, customer geolocation, customers GMT time. Rank the advertisement which is often viewed by the customer

2.2 Experimental Setup

Fig 1.Represent the admin in which he can posts the advertisement and to regulate the process on how many times the user have scrolled the advertisement can be viewed by the admin and user is the customer so he can scroll the advertisement and see that in chance they can purchase the product

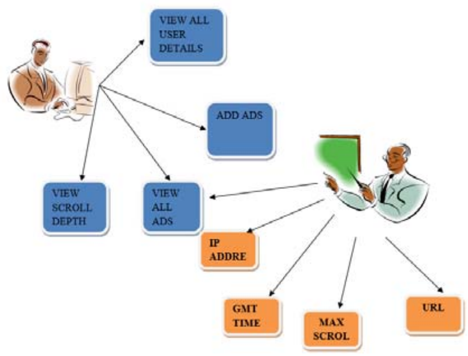


Figure 1: Architectural Diagram of Ad view ability

By using this proposed system we can use the scroll depth of the user ip address of the user url of the web page and user geolocation user time gmt time.

Re- ranking algorithm the advertisement that is frequently used by the user

3. PLC CONST: PREDICTION MODEL WITH CONSTANT MEMBERSHIPS

Our 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 device type and geo-location are easy to be modeled.

However, some other significant features are very hard to capture due to lack of data and the ambiguity of user webpage interaction. For example, pages with popular

To overcome these issues, we use a latent class model [25] to discover classes of users and webpages.

Specifically, we build a probabilistic latent class model with constant memberships (PLC const). The intuition behind it is that different latent classes of webpages and users tend to generate different levels of max scroll depths. PLC const can detect classes of users and webpages that share similar patterns of max scroll depth. The exact class memberships of each user and webpage are learnt from the user log and used to do prediction for each page view in test datasets. PLC const outputs the probability $P(x_{ua}|u,a)$, where x_{ua} is the max scroll depth that a user u reaches on a page a. Formally, PLC const works as follow:

$$N_s N_p P(x_{ua}|u,a) = {}^{XX}P(s|u)P(p|a)P(x_{ua}|f^{uac},s,p;w_{sp}) \quad (1)$$

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. $P(s|u)$ is the probability that user u belongs to the latent user class s, while $P(p|a)$ is the probability that webpage a belongs to the latent webpage class p. For simplicity, in this paper, we use s and p to notate individual latent user classes and latent page classes. The last term, $P(x_{ua}|f^{uac},s,p;w_{sp})$, represents the probability that the max scroll depth of the page view is x_{ua}, given the latent user class s and webpage class p. f^{uac} is the feature set that reflects the user, the webpage, and context information, while w_{sp} is the corresponding feature weights.

As mentioned above, the last term can be approximated by the probability density function of a normal distribution. Note that there is no single distribution that can fit all datasets. This paper proposes a general framework for predicting user reading behavior. The proposed methods do not rely on properties specific to the Gaussian distribution. Therefore, different publishers and advertisers can plug in other distributions according to their own datasets. They only need to change the probability density function (Equation 2) and the corresponding M-step.

$$P(x_{ua}|f^{uac},s,p;w_{sp}) = \frac{1}{\sqrt{2\pi\sigma_{sp}^2}} \cdot \exp\left(-\frac{(x_{ua}-w_{sp}^T\cdot f^{uac})^2}{2\sigma_{sp}^2}\right) \quad (2)$$

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., $\mu_{ua} = w_{sp}^T\cdot f^{uac}$. The superscript *uac* means the feature set includes user, webpage, and context features. Each pair of latent user class s and latent webpage class p has a set of w_{sp*}, i.e., the weights in the linear function of μ_{ua} and σ_{sp}, i.e., the mean and the standard deviation.

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:

1. The mean max scroll depth of a by all users. This feature captures the popularity of the webpage.
2. 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:

Interaction of the mean max scroll depth of u and that of a, i.e., the product of features 1 and 3.

Page View Context:

User geo-locations, which were shown to be important by our analysis of the dataset. 7) Device Type (i.e., desktop, mobile, or tablet), also shown to have a certain relevance by our analysis.

Let **W** be the collection of the weight vectors of all latent user classes and webpage classes. σ is the collection of the standard deviations of all latent user classes and webpage classes. The features help iteratively determine **W** and σ.

In Equations 1 and 2, there are several parameters ($P(s|u)$, $P(p|a)$, **W**, σ). They can be calculated by maximizing the following likelihood function:

$$l(P(s|u),P(p|a),\mathbf{W},\sigma) = \sum_{u,a} \ln\left(\sum^{N_s}\sum^{N_p} P(s|u)P(p|a)P(x_{ua}|f_{uac},s,p;w_{sp})\right) \quad (3)$$

To maximize it, the Expectation Maximization (EM) Algorithm is adopted. The EM algorithm is widely used to solve the maximum-likelihood parameter estimation problem. The EM algorithm performs an expectation step (Estep) 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.

$$P(s,p|f^{uac};w_{sp}) = P(s|u)P(p|a)\cdot\frac{1}{\sqrt{2\pi\sigma_{sp}^2}}\cdot \exp\left(-\frac{(x_{ua}-w_{sp}^T f^{uac})^2}{2\sigma_{sp}^2}\right) \quad (4)$$

The M-step updates the parameters in Equation 4, which can maximize Equation 3. In each iteration, the M-step updates the value of each parameter based on the result of the E-step. The updated $w_{sp}{}^*$ of each iteration in Equation 7 can be determined by Limited-memory BFGS, an optimization algorithm in the family of quasi-Newton methods.

$$P(s|u)^* \propto \sum_{p,a} P(s,p|f^{uac}) \quad (5)$$

$$P(p|a)^* \propto \sum_{s,u} P(s,p|f^{uac}) \quad (6)$$

$$\dot{w}_{sp*} \propto \arg\max_{w_{sp}}\{-\sum_{u,a} P(s|u)P(p|a)\cdot \left[\frac{(x_{ua}-w_{sp}^T f^{uac})^2}{2\sigma_{sp}^2} + \ln\sigma_{sp} + \ln\sqrt{2\pi}\right]\} \quad (7)$$

$$\sigma_{sp}^* \propto \sqrt{\frac{\sum_{ua} P(s|u)P(p|a)(x_{ua}-w_{sp}^T f^{uac})^2}{\sum_{ua} P(s|u)P(p|a)}} \quad (8)$$

The EM iterations stop if the max ratio is not greater than a pre-defined threshold, which is set to 10^{-3} in our experiments. In other words, it stops if the difference of all feature weights is less than 10^{-3}.

After convergence, the PLC const with the optimal parameters can predict $P(x_{ua}|u,a)$, i.e., the probability density of any target max scroll depth x_{ua} of a user-webpage pair. Section 7 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.

4. PLC DYN: PREDICTION MODEL WITH DYNAMIC MEMBERSHIPS

By computing offline the memberships of users and webpages belonging to latent user and webpage classes, PLC const predicts the viewability of any target scroll depth in a page view. However, user and webpage memberships in reality can be dynamic during the online process, since user interests and page popularity keep changing. For instance, user interests may shift over time, e.g. from entertainment to sports, which can influence the class memberships of a user. Webpage attractiveness may also change for some reasons, e.g., bursting topics and content freshness. For instance, users viewing a newly updated webpage may scroll deeper than users viewing the same page one week later. The reason is that after one week its content is not fresh and attractive. A drawback of PLC const is that it can only use fixed memberships calculated from training data to make predictions in test data. For instance, assuming there are two user classes, the memberships of a user in the training data are $s_1 = 0.8$ and $s_2 = 0.2$, i.e., the probability that the user belongs to the first latent user class is 0.8. These memberships are used to predict in all test page views involving that user. Thus, PLC const cannot adapt user's interest shift.

To capture the dynamic nature of the memberships, we propose to represent the memberships by a function whose output value is determined in real-time.

Meanwhile, the feature vectors should also be able to reflect the change of user, webpage, and context. Based on this idea, we develop a dynamic probabilistic latent class model, PLC dyn that extends PLC const. This model enables dynamic memberships and also considers webpage information, such as channels, i.e., topical categories (e.g. "finance" and "lifestyle"), and sections, i.e., sub-channels. Webpage information is provided by the article metadata. Note that "dynamic" does not refer to online learning where the model parameters keeps changing based on incoming data stream. The model parameters, i.e. feature weights, are not changed during testing once they have been learnt from the training data. But the memberships calculated based on the model parameters are dynamically changing since feature values may change over time.

Let us clarify the similarities and differences between PLC const and PLC dyn in technical details. Similar with PLC const, PLC dyn calculates the probability that a user or a page belongs to each class and utilizes user and webpage classes to overcome sparsity. However, unlike PLC const, PLC dyn calculates the user and page memberships in real-time, instead of learning constant numbers of memberships from training data offline. In particular, in Equation 1, the memberships $P(s|u)$ and $P(p|a)$ are constant numbers learnt from training data. Before being retrained, PLC const always uses these fixed memberships to perform predictions for specific users and pages. In contrast, PLC dyn uses soft-max functions powered by linear functions to calculate user and webpage memberships, as shown in Equation 10. PLC dyn learns the feature weights in the linear functions from training data, rather than learning final memberships. These feature weights are used to compute the memberships in real-time with the feature values at that moment. Thus, the memberships of a user or a page may be different over time, i.e. dynamic, since the feature values keep updating. For instance, the value of the feature "the mean max scroll depth of the user on the webpages in the same section" is dynamic. It can capture the change of the user's interest. Also, the dynamic value of the feature "the mean max scroll depth of the pages in the same section" can capture the change of topic attractiveness. Thus, PLC dyn can better adapt to changes of user and page characteristics. To support such calculation, user features and webpage features (webpage features are not used in PLC const) are used to calculate the user and page memberships, respectively.

Formally, PLC dyn is modeled as following.

$P(x_{ua}|u,a)$ =*Ns Np* (9)

XX$P(s|fu;\alpha\ ) \cdot P(p|fa;\beta p)P(xua|fuac,s,p;wsp)\ s$

where $P(s|f^u;\alpha_s)$ represents the probability that the user u with the user features f^u and the corresponding feature weights α_s belongs to the latent user class s, while the $P(p|f^a;\beta_p)$ represents the probability that the webpage a with the webpage features f^a and the weights β_p belongs to the latent webpage class p. $P(x_{ua}|f^{uac},s,p;w_{sp})$ is the probability that the max scroll depth is x_{ua} given the user and the webpage belonging to s and p respectively. It is almost the same as its counterpart in Equation 1, but they have different feature vectors. f^{uac} is the entire feature set that concatenates all features about the user, the webpage, and the context (e.g. screen sizes, devices), while w_{sp} is the corresponding feature weights. N_s and N_p are the numbers of latent user and webpage classes, respectively.

Equation 9 uses user features f^u and webpage features f^a to calculate the user and webpage memberships, respectively. The parameters that have to be learnt from the training data are feature weights α_s and β_p. In contrast, the memberships in Equation 1 are learnt as constant numbers, $P(s|u)$ and $P(p|a)$. Each user and each webpage receives a set of membership values, which are not subject to change during prediction.

The user membership $P(s|f^u;\alpha_s)$ and the webpage membership $P(p|f^a;\beta_p)$ can be modeled by the soft-max function [26]. The soft-max function takes the outcome of a linear function as input and outputs the predicted probability for one of the classes given the input vector. User and webpage memberships can be defined:

$$P(s|f^u;\alpha_s) = \frac{1}{Z_u}\exp(\alpha_s^T f^u) = \frac{\exp(\alpha_s^T f^u)}{\sum^{N_s}\exp(\alpha_s^T f^u)} \quad (10)$$

where Z_u is the normalization factor that guarantees the sum of the memberships of a user belonging to all classes is equal to one. The page membership with weights β_p and page features f^a can be modeled similarly. As in PLC const, the last term can be modeled by Equation 9:

$$P(x_{ua}|f^{uac},s,p;w_{sp}) = \frac{1}{\sqrt{2\pi\sigma_{sp}^2}}\exp\left(\frac{(x_{ua}-w_{sp}^T f^{uac})^2}{-2\sigma_{sp}^2}\right. \quad (11)$$

f^{uac} is the combination of the user , webpage, and context features. All features are shown as below.

User Features (f^u):

1) The mean max scroll depth of the user in past page views, which captures user browsing habits. 2) The mean scroll depth of the user on the webpages in the same channel. 3) The availability of the second feature. 4) The mean scroll depth of the user on the webpages in the same section, i.e., sub-channel. 5) The availability of the fourth feature. 6) The mean scroll depth of the users at the same geo location on the webpages in the same channel. 7) The mean scroll depth of the users at the same geo location on the webpages in the same section.

Webpage Features (f^a):

1. The mean max scroll depth of the page by all users.
2. This feature captures the popularity of the webpage. 2) The mean max scroll depth of the pages in the same channel, i.e., topical category (e.g., finance).
3. The mean max scroll depth of the pages in the same section, i.e. sub-channel.
4. The mean max scroll depth of all pages in the "related content list" of the page. If the page has no related content page, it equals to the first feature. 5) The length of the body text.

Page View Context (f^c):

Screen Width, i.e., the width of the user's screen. 2) Screen Height

Viewport Width, i.e., the viewport is the visible area of a web page on user's screen. Unlike screen size, viewport size indicates the area of the user's browser. Viewport size is captured and sent to the server when the user clicks the link of the page.

Viewport Height.
The mean max scroll depth of all page views on the same device.

Note that only the first user feature and the first webpage feature are used in both PLC const and PLC dyn. Other features are either new features added in PLC dyn (e.g., screen size and the mean max scroll depth of the pages in the same channel) or the dynamic version of the features used in PLC const (e.g., the mean max scroll depth of all page views on the same devices). Also, since the user and webpage characteristics can be reflected in their own memberships, the interaction used in PLC const is removed.

The new feature set contains many categorical characteristics, e.g., channels, sections, and geo-locations. To reduce the number of dimensions and enable dynamic updates, these categorical characteristics are converted to continuous features. For instance, we convert "device type" (used in the PLC const) to "the mean max scroll depth of all page views on the same devices" (used in PLC dyn). Specifically, in PLC dyn, the continuous variable "the mean max scroll depth of all page views on the same device" is adopted, instead of dummy variables representing devices. This feature in PLC dyn occupies only one dimension, while its counterpart feature in PLC const has three dimensions. In addition, the value of PLC dyns feature is dynamic, since the mean scroll depth is changing over time. In contrast, being represented by dummy variables, the value of PLC consts feature is constant.

(α,β,W,σ) denote the weight vectors of all latent user and webpage classes as well as the weight vectors and standard deviations of all latent user and webpage class pairs, respectively. These parameters can be learnt by maximizing the following likelihood function. Note that the differences between Equations 12 and 3 are the same as those between Equations 9 and 1. $l(\alpha,\beta,W,\sigma) =$

$$\sum_{u,a} \ln \left(\sum^{N_s} \sum^{N_p} P(s|f^u;\alpha_s) P(p|f^a \beta_p) P(x_{ua}|f^{uac}, s, p; w_{sp}) \right) \quad (12)$$

Similar with PLC const, the EM algorithm is adopted to learn the parameters iteratively in PLC dyn. The E-step is as below:

$$P(s,p|f^{uac};w_{sp}) = \frac{P(s|f^u;\alpha_s)P(p|f^a\beta_p)P(x_{ua}|f^{uac},s,p;w_{sp})}{\sum^{N_s N_p} P(s|f^u;\alpha_s)P(p|f^a\beta_p)P(x_{ua}|f^{uac},s,p;w_{sp})} \quad (13)$$

The values of the parameters are updated in the corresponding M-step using the L-BFGS algorithm:

$$\alpha^*_{s\cdot} \propto \underset{\alpha_{s\cdot}}{argmax} \sum_{u,a} \left[\sum_{p} P(s,p|f^{ua}) \right] \cdot \ln \left[\frac{1}{Z_u} \cdot \exp\left(\alpha_s^T f^u\right) \right] - \frac{\lambda}{2} \alpha_s^2 \quad (14)$$

$$\beta^*_{p\cdot} \propto \underset{\beta_{pj\cdot}}{argmax} \sum_{u,a} \left[\sum_{s} P(s,p|f^{ua}) \right] \cdot \ln \left[\frac{1}{Z_a} \cdot \exp\left(\beta_p^T f^a\right) \right] - \frac{\lambda}{2} \beta_p^2 \quad (15)$$

$$w^*_{sp\cdot} \propto \underset{w_{sp\cdot}}{argmax} \sum_{u,a} P(s,p|f^{ua}) \cdot \ln \left[\frac{1}{\sqrt{2\pi\sigma_{sp}^2}} \cdot \exp\left(\frac{\left(x_{ua} - w_{sp}^T f^{uac}\right)^2}{-2\sigma_{sp}^2} \right) \right] \quad (16)$$

$$\sigma^*_{sp} \propto \underset{\sigma_{sp}}{argmax} \sum_{u,a} P(s,p|f^{ua}) \cdot \ln \left[\frac{1}{\sqrt{2\pi\sigma_{sp}^2}} \cdot \exp\left(\frac{\left(x_{ua} - w_{sp}^T f^{uac}\right)^2}{-2\sigma_{sp}^2} \right) \right] \quad (17)$$

Note that the first terms of Equation 14 and 15 are not strictly convex. Therefore, adding weight decay, i.e., the second terms, will take care of the numerical problems associated with soft-max regression's over-parametrized representation. The second terms penalize large values of the parameters, α and β, and thus guarantee to have a unique solution, i.e., converge to the global maximum. λ is the weight decay term, which should be greater than 0. In the experiment, it is set to 0.01 based on cross validation. After convergence, the PLC models with the optimal parameters can predict $P(x_{ua}|u,a)$, i.e., the probability density of any target max scroll depth x_{ua} of a user-webpage pair.

5. VIEWABILITY PREDICTION FOR A TARGET SCROLL DEPTH

Given a target scroll depth X and a user-webpage pair, the trained PLC models can be used to compute the probability that the max scroll depth will be X, i.e., $P(x_{ua} = X|u,a)$. As stated in the problem definition, our goal 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 18. 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 page bottom.

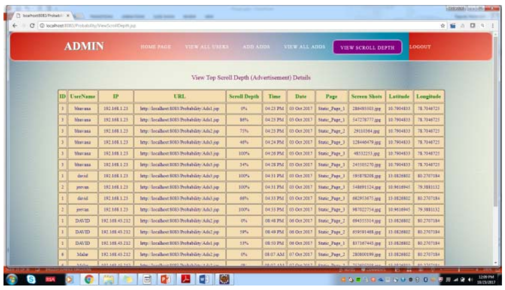
$$P(x_{ua} \geq X|u,a) = \int_X^{100\%} P(x_{ua}|u,a)dx_{ua} \quad (18)$$

6. RESULTS AND DISCUSSIONS

On Various analysis, we have done on the field of advertisement. We have proposed the following results on the basis of the investors to improve the view ability of the advertisement.

In Fig 2. Represents the scroll percentage is there to identify the total analysis of the customer this scroll percentage from user scroll the mouse that value based predicts. Admin should have maintained to all users and user view all ads to maintain.this analysis can be viewed on the admin side

Figure 2: View Scroll Percentage



In Fig 3. Represents the top scroll depth is for the most viewed advertisement that comes in the top place and this helps the investors to identify the particular product and they can improve the purchase of the customer they can be viewed on the admin side

Figure 3: Top Scroll Depth

In Fig 4. Represents the scroll percentage can be viewed even in the customer side also that is how many times the customer have scrolled the product and viewed can be seen in this so this may also create an Impact on the user side

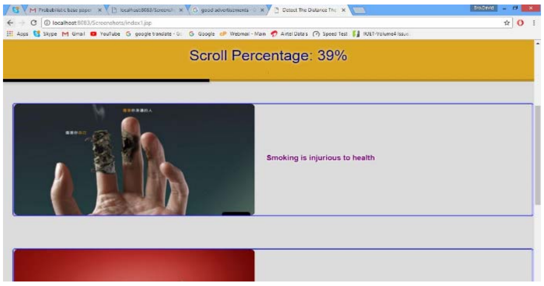


Figure 4: Scroll Percentage

7. CONCLUSION AND FUTURE WORK

Therefore by using the proposed system, by using scroll depth process we can able to view the predictability of the advertisement based on scroll percentage and also predict the maximum of viewed advertisement. In Future we can improve the rate of online shopping of the customers and to improve the benefits of online advertisers to purchase the products from the online so this is a mutual concern for the investors and the customers

REFERENCES

[1] I.Lunden,"Internet ad spend to reach $121b in 2014," http://techcrunch.com/2014/04/07/internet-ad-spend-to-reach121b-in-2014-23-of-537b-total-ad-spend-ad-tech-gives-display-aboost-over-search/.

[2] Y. Chen and T. W. Yan, "Position-normalized click prediction in search advertising," in *KDD'12*, 2012, pp. 795–803.

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

[4] 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*, 2014, pp. 39–56.

[5] Google, "The importance of being seen," http://think.storage.googleapis.com/docs/the-importanceof-being-seen study.pdf.

[6] M. Mareck, "Is online audience measurement coming of age?" *Research World*, vol. 2015, no. 51, pp. 16–19, 2015.

[7] 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.

[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*, 2012, p. 9.

[9] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer, "Not quite the average: An empirical study of web use," *ACM TWEB*, vol. 2, no. 1, p. 5, 2008.

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

[11] M. Holub and M. Bielikova, "Estimation of user interest in visited web page," in WWW'10, 2010, pp. 1111–1112.

[12] C.-J. Wang and H.-H. Chen, "Learning user behaviors for advertisements click prediction," in ACM SIGIR'11 Workshop on Internet Advertising, 2011, pp. 1–6.

[13] Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by E. incorporating user behavior information," in *ACM*

[14] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM TIST*, vol. 5, no. 4, p. 61, 2014.

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

[16] C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey, "Click-through prediction for advertising in twitter timeline," in *In Proceedings of KDD'15*. ACM, 2015, pp. 1959–1968.

[17] C. Liu, R. W. White, and S. Dumais, "Understanding web browsing behaviors through weibull analysis of dwell time," in *ACM SIGIR'10*, 2010, pp. 379–386. X. Yi, L. Hong, E. Zhong, N. N. Liu, and S. Rajan, "Beyond clicks:

[18] dwell time for personalization," in *Recsys'15*, 2014, pp. 113–120.

[19] P. Yin, P. Luo, W.-C. Lee, and M. Wang, "Silence is also evidence: interpreting dwell time for recommendation from psychological perspective," in *KDD'13*. ACM, 2013, pp. 989–997.

[20] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. Springer, 2011.

[21] M. Zanker, "A collaborative constraint-based meta-level recommender," in *In Proceedings of RecSys'08*. ACM, 2008, pp. 139–146.

[22] J. Sun, S. Wang, B. J. Gao, and J. Ma, "Learning to rank for hybrid recommendation," in *CIKM'12*.ACM, 2012, pp. 2239–2242.

[23] D.-P. C., "Lifelines," https://github.com/camdavidsonpilon/lifelines, 2016.

[24] S. Rendle, "Factorization machines with libfm," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 3, no. 3, p. 57, 2012.

[25] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE TKDE, vol. 21, no. 9, pp. 1263–1284, 2009.

[26] S. Cetintas, L. Si, Y. P. Xin, and R. Tzur, "Probabilistic latent class models for predicting student performance," in In Proceedings of CIKM'13. ACM, 2013, pp. 1513–1516.