Predicting Bounce Rates in Sponsored Search Advertisements

D. Sculley, Robert Malkin, Sugato Basu, Roberto J. Bayardo
Google, Inc.
{dsculley, rgm, sugato}@google.com, bayardo@alum.mit.edu

ABSTRACT

This paper explores an important and relatively unstudied quality measure of a sponsored search advertisement: bounce rate. The *bounce rate* of an ad can be informally defined as the fraction of users who click on the ad but almost immediately move on to other tasks. A high bounce rate can lead to poor advertiser return on investment, and suggests search engine users may be having a poor experience following the click. In this paper, we first provide quantitative analysis showing that bounce rate is an effective measure of user satisfaction. We then address the question, *can we predict bounce rate by analyzing the features of the advertisement?* An affirmative answer would allow advertisers and search engines to predict the effectiveness and quality of advertisements before they are shown. We propose solutions to this problem involving large-scale learning methods that leverage features drawn from ad creatives in addition to their keywords and landing pages.

Categories and Subject Descriptors

I.5.4 [**Computing Methodologies**]: Pattern Recognition—*Applications*

General Terms

Experimentation

Keywords

Bounce Rate, Sponsored Search, Machine Learning

1. INTRODUCTION: BOUNCE RATE

Sponsored search advertising allows advertisers to measure and monitor their return on investment with an unprecedented level accuracy and detail. There are several performance metrics advertisers use to monitor the effectiveness of their advertising campaigns, and many of these metrics are also useful to search engine providers who aim to provide users with search advertising that is both relevant and useful. Among the best known metrics for these purposes is *click through rate* (CTR) and *conversion rate* (CvR). Though less studied, another important metric of advertising effectiveness is *bounce rate*, which Avinash Kaushik of Google Analytics colorfully describes as follows: [17, 18]:

> Bounce rate for a page is the number of people who entered the site on a page and left right away. They came, they said yuk and they were on their way.

Kaushik claims bounce rate is important for advertisers to monitor because a user who bounces from a site is unlikely to perform a conversion action such as a purchase. He suggests that high bounce rates may indicate that users are dissatisfied with page content or layout, or that the page is not well aligned to their original query. Although bounce rate is an intuitive and widely-used metric, it has not been extensively studied. To our knowledge, this paper is the first formal investigation of bounce rate in the literature.

To better understand the nature of bounce rates in sponsored search advertisements, we devote the first part of this paper to answering the following fundamental questions with large-scale data analysis:

- Can we quantify the intuition that bounce rate is an effective measure of user satisfaction?
- How do bounce rates vary by page language or search query?

We provide quantitative and qualitative answers to these questions in Section 3, following formal definitions of bounce rates and a discussion of observing bounce rates with anonymized and non-invasive methods given in Section 2.

Unfortunately, empirically measuring the bounce rate of an ad requires significant click history. Depending on the click through rate of the ad, it may require hundreds if not thousands of impressions before its bounce rate can be accurately estimated. The following question thus motivates the second portion of this paper:

- Can we effectively predict bounce rates in the absence of significant historical observation?

An accurate prediction model for bounce rates could allow advertisers to determine an ad's effectiveness with fewer clicks. Advertisers can use such knowledge to lower their costs and improve return on investment by more quickly

KDD'09, June 28–July 1, 2009, Paris, France.

shutting down ads that are likely to perform poorly. Additionally, it could be used by the advertiser to guide the up front creation of their advertisements and landing pages. Finally, the model might also help sponsored search systems quickly estimate user satisfaction of candidate ads.

We tackle this challenge in Sections 4 and 5, in which we apply two large-scale machine learning approaches and test a range of feature types for predicting bounce rates. We give results both for the task of predicting bounce rates on new (unseen) advertisements and for populations of advertisements over time; additionally, we provide detailed analysis of the impact of various feature types. As discussed in Section 6, this machine learning approach to bounce rate prediction is motivated by prior success in predicting CTR for sponsored search, as exemplified by Richardson *et al.* [28].

2. BACKGROUND

This section provides a brief background on sponsored search, gives a formal definition of bounce rate, and discusses methods for observing bounce rate non-intrusively.

2.1 Sponsored Search Terminology

Sponsored search is the problem of delivering advertisements in response to search queries on an internet search engine. In the sponsored search setting, a search engine *user* submits a *query*, such as `flat screen television`. In response to this query, the search engine displays a set of *algorithmic search results* (which are not influenced by advertisers) and a set of advertiser-provided *sponsored advertisements*. A sponsored advertisement, or *ad* for short, consists of *creative text*, which is a brief (e.g., 3 line) description to be displayed by the search engine, a *keyword*, which specifies the query for which the creative is eligible for display, and a *landing page* specifying the click destination (such as a page where the user may buy flat screen televisions). An ad *impression* results when the search engine displays the ad in response to a user query, and a *clickthrough* results when the user viewing the ad clicks on it and and visits the ad's landing page. In most sponsored search settings, advertisers pay a small fee to the search engine for each user click, with the price determined by an auction system. The search engine attempts to select and rank ads in a manner such that they are relevant and useful to the search engine user.

2.2 Defining Bounce Rate

Recall that, informally, the bounce rate of an advertisement is the fraction of users who click on the ad and immediately move on to other tasks (or *bounce* from the site). To formalize this concept, we must have a bounce threshold Θ defined, and define the *bounce rate* of an advertisement as the fraction of all clickthroughs that result in a bounce within time Θ. Typical values for Θ range from 5 to 60 seconds; our preliminary analysis showed qualitatively similar results for a range of values. Within this paper, we use the same fixed Θ for all experiments.

2.3 Measuring Bounce Rate

It is easy for advertisers to measure bounce rates of their advertisements in an anonymous, non-invasive manner, using aggregate data from the webserver hosting the advertiser's site. An advertiser might designate a clickthrough as a bounce if it fails to generate new observable events after time Θ following the clickthrough. This methodology may incur some false positives, e.g., consider a user who calls the advertiser rather than continuing to navigate the site.

A search engine provider, on the other hand, can observe a user's behavior on the search engine itself, but cannot make observations after the user has clicked through to an advertisement. In this situation, a clickthrough might be classified as a bounce if other events from the user are observed on the search engine within a time Θ following the clickthrough. Again, a small number of false positives are possible for a variety of reasons.

Clearly, any practical method of observing user bounces is prone to some error. The bounce rates we obtain from observations are therefore estimates of the true bounce rate. We expect that such observations lead to estimates that are strongly correlated with true bounce rate, particularly when observations are made over a large number of clickthroughs.

3. ANALYSIS OF BOUNCE RATE

In this section, we explore a set of fundamental qualities connected with bounce rate. The goals of this section are to quantify the intuition that bounce rate is an effective measure of user satisfaction and to examine the factors that cause bounce rates to vary.

3.1 Normalization of Reported Metrics

All values of user-based metrics reported in this paper (bounce rate and click-through rate) have been been preprocessed as follows, to protect privacy while allowing repeatability. Values in the upper and lower deciles of the metric were removed from the data set to eliminate outliers, and then the remaining values were normalized by the difference between the remaining maximum and minimum value. This results in each metric being rescaled to the range [0,1]. Popularity metrics for languages, keywords, and categories were computed similarly, but using the natural log of the raw frequencies.

3.2 Bounce Rate and Click Through Rate

As described above, one traditional observable measure of user satisfaction with an ad is its click through rate (CTR) [28]. This is defined as:

$$CTR = \frac{\text{total clicks on the ad}}{\text{total ad impressions}}$$

CTR naturally captures the advertisement's relevance as perceived by search engine users. Users who think the advertisement is relevant to their search will be more likely to click on that advertisement.

One limitation of CTR is that it fails to capture the user's evaluation of the ensuing experience on the landing page because the landing page is not visible prior to the click; the only visible information prior to the click is the creative text. In contrast, bounce rate measures the impact of the landing page on the user's behavior.

We compare bounce rate to CTR in Figure 1, which plots observed bounce rate to observed CTR for several million sponsored search advertisements shown in response to queries on google.com during a recent month. This figure shows a striking trend: advertisements with very low observed bounce rate have very high CTR. The Pearson correlation coefficient between bounce rate and CTR is -0.85, indicating a strong inverse relationship. Thus, if CTR is a good proxy for user satisfaction, then bounce rate is also a good proxy

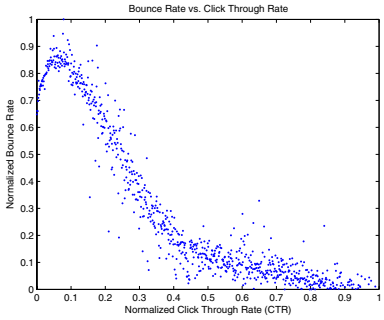

Figure 1: Comparing bounce rates with CTR.

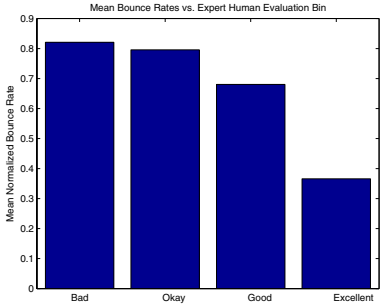

Figure 2: Comparing bounce rates with expert human evaluations.

for user satisfaction. This correlation is interesting, since CTR is based on the user's evaluation of creative text while bounce rate depends more on the user's evaluation of the landing page. This correlation is not spurious; for contrast, there is almost no correlation (.003) between an ad's bounce rate and its maximum click cost.

There are a few possible interpretations of this correlation. One interpretation is that advertisers who target their advertisements carefully also tend to provide goods, services, or information resulting in high user satisfaction. Another interpretation is that certain advertisers develop good reputations among users, resulting in a higher likelihood to click and a lower likelihood of bouncing. A final possibility is that some users are able to infer the quality of the landing page from the creative text. For example, some users may suspect that creatives promising `Make free money fast!` will be less satisfactory than a creative promoting a `Professional, licensed tax accountant`.

3.3 Bounce Rate and Expert Evaluation

To further quantify our intuition that bounce rate is an effective measure of user satisfaction with sponsored search advertisements, we gathered expert human evaluation for a sample of 7,000 advertisements shown on Google Search in a recent month. These advertisements were randomly sampled from all advertisements shown, weighted by impressions. We sampled advertisements in English, Spanish, and six other languages from continental Europe and Asia.

The advertisements in this sample were rated by expert human evaluators, who judged each advertisement and rated it for quality based on expected user satisfaction into one of four ratings: `excellent`, `good`, `okay`, and `bad`. The evaluators were given no information about the bounce rates, CTR, or other predictive information; their judgements were based solely on the contents of the advertisement landing page.

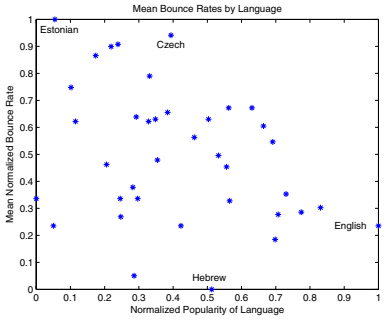

Figure 3: Mean bounce rates by language.

The results, shown in Figure 2, show that expert human evaluation of advertisement quality agree well with implied user assessment given by bounce rate. The normalized mean bounce rates for ads in the `excellent` category were less than half that of those in the `bad` category, and there is a monotonic decrease in bounce rate in each successive rating bin from `bad` to `excellent`. So, bounce rate is well correlated with expert opinion.

3.4 Bounce Rate and Quality Guidelines

Continuing in this vein, we examined the connection between mean bounce rates and the suggested quality guidelines for advertisers in Google Search.[1] These include suggestions regarding style and technical requirements of ad text, guidelines about content, and recommendations to help advertisers ensure their landing pages will satisfy users.

We collected a few thousand advertisements and had expert human evaluators flag advertisements violating one or more of the guidelines. The mean bounce rate for the advertisements that followed the guidelines was 25.4% lesser than the mean bounce rate for the sample of advertisements that did not follow these guidelines. A t-test measured this difference in sample means as significant with $p < 0.0001$.

3.5 Distribution of Bounce Rates

So far we have seen that bounce rate is a good measure of observed user satisfaction by a strong correlation to CTR, and is a good indicator of expected user experience via the connection to expert human evaluation and quality guidelines. To further increase the reader's intuition regarding this possibly unfamiliar metric, we explore the distribution of bounce rates in different groupings: advertisement language and query keyword.

Bounce Rates by Language.

We measured observed mean bounce rates for all ads shown on Google Search in a recent month across 40 languages. These normalized mean bounce rates are plotted against a "popularity" score, as described in Section 3.1. These results are shown in Figure 3.

We observe that mean bounce rates vary significantly by language. One possible explanation for this observation may involve market maturity. English language advertisements (the most popular by our data) have a relatively low bounce rates, while languages representing emerging markets have significantly higher bounce rates. Whether this is because advertisers in established markets produce higher quality

[1]http://adwords.google.com/support/bin/static.py?page=guidelines.cs&topic=9271&view=all

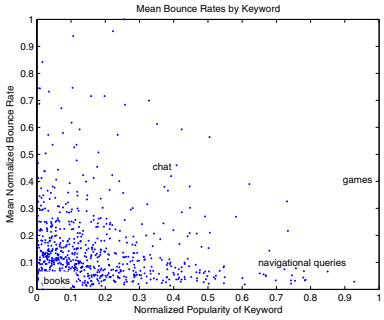

Figure 4: Mean bounce rates by keyword.

advertisements, or because users in emerging markets are less receptive to online advertisement, is an open question.

Bounce Rates by Keyword.

We also examined the mean bounce rate for 749 keywords, drawn from a random sample of the multiset of all keywords on Google Search in a recent month. We plotted mean bounce rate for each keyword against its popularity score, and show the results in Figure 4.

This figure shows mean bounce rates vary significantly by particular keyword. Navigational queries, such as those containing specific business names, result in very low bounce rates. Commercial terms, such as `books` and `flights`, also have low bounce rates. Entertainment oriented terms such as `games` and `chat` exhibited much higher bounce rates.

In general, there is a rough inverse relationship between keyword popularity and mean bounce rate for that keyword. This may be because the greater competition for these more popular keywords creates a need for these competing advertisers to achieve higher standards of quality.

4. PREDICTING BOUNCE RATE

Our primary aim in this paper is to predict the bounce rate of an ad with little to no click history. Formally, we represent an ad as a triple $(\mathbf{q}, \mathbf{c}, \mathbf{p})$ consisting respectively of its keyword, creative, and landing page. Our goal is to predict a bounce rate close to the true bounce rate $B_{rate}(\mathbf{q}, \mathbf{c}, \mathbf{p})$. Since the true bounce rate has a range of [0,1], this prediction problem fits naturally within a regression framework.

Logistic regression [3] or support vector machine (SVM) regression [15] with probability estimation [25] for bounce rate prediction requires a mapping $\mathbf{x}(\cdot,\cdot,\cdot) \mapsto \mathbb{R}^n$ from a query, creative, landing page triple to an n dimensional *feature vector*. The feature mapping explored in this paper, as detailed in Section 4.3, results in a high dimensional space comprised of millions of features. Furthermore, our training data sets contain millions of examples. We now review two methods for dealing with such large data sets: parallelized logistic regression, and ϵ-accurate SVM regression.

4.1 Logistic Regression

A (binary) logistic regression model consists of a *weight vector* $\mathbf{w} \in \mathbb{R}^n$ which is used to make predictions $f(\cdot)$ on the features $\mathbf{x} \in \mathbb{R}^n$ of a data example using a sigmoid function:

$$\Pr(y_{\mathbf{x}} = 1|\mathbf{x}, \mathbf{w}) = f(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}\cdot\mathbf{x})}}$$

where $\mathbf{w} \cdot \mathbf{x}$ is the dot product of the weight vector $\mathbf{w}$ and the input vector $\mathbf{x}$, and $y_{\mathbf{x}}$ is the label on $\mathbf{x}$ that indicates a bounce event. Logistic regression is often used in cases where the expected output is a probability, as its output is naturally constrained to the range [0,1].

The weight vector $\mathbf{w}$ is estimated by maximizing the log likelihood of f over a set of training data S. Because many features may be irrelevant, we encourage sparsity using regularization based on an L1-norm penalty on the weight vector:

$$\max_{\mathbf{w}} \sum_{\mathbf{x}\in S} \log \Pr(y_{\mathbf{x}}|\mathbf{x}, \mathbf{w}) - \lambda||\mathbf{w}||_1, \quad (1)$$

where $y_{\mathbf{x}}$ is the true label for example $\mathbf{x}$, $||\mathbf{w}||_1$ is the L1-norm of $\mathbf{w}$, and λ is a regularization term setting the level of importance to place on finding a sparse model versus finding a model that minimizes predictive error. In applications of logistic regression for classification, $y_{\mathbf{x}}$ is often drawn from the binary set $\{0, 1\}$. If the dataset has $c_1(\mathbf{x})$ instances of $\mathbf{x}$ with $y_{\mathbf{x}} = 1$ and $c_0(\mathbf{x})$ instances of $\mathbf{x}$ with $y_{\mathbf{x}} = 0$, Equation 1 can be equivalently written as:

$$\max_{\mathbf{w}} \sum_{\text{unique } \mathbf{x}\in S} c_y(\mathbf{x})\big[y \log\big(\Pr(y|\mathbf{x}, \mathbf{w})\big) + (1-y)\log\big(1 - \Pr(y|\mathbf{x}, \mathbf{w})\big)\big] - \lambda||\mathbf{w}||_1, \quad (2)$$

where y is 0 or 1. In our application we observe the bounce rate $B_{rate}(\mathbf{x}) \in [0, 1]$ of an ad with feature vector $\mathbf{x}$. We optimize Equation 2 setting $c_1(\mathbf{x}) = B_{rate}(\mathbf{x})$ and $c_0(\mathbf{x}) = 1 - B_{rate}(\mathbf{x})$. Note that this is equivalent to considering $kB_{rate}(\mathbf{x})$ examples of $\mathbf{x}$ with $y_{\mathbf{x}} = 1$ and $k(1 - B_{rate}(\mathbf{x}))$ examples of $\mathbf{x}$ with $y_{\mathbf{x}} = 0$, where k is a scaling constant.

This optimization problem may be solved via methods such as LBFGS [21]; however, for very large data sets, these methods do not scale well due to large matrix manipulations. We thus use stochastic gradient descent as a viable alternative [19], noting that the non-differentiability induced by the L1 penalty term can be handled by methods similar to truncated gradient descent [20]. To achieve scalability, we use a parallelized learning algorithm where each machine handles a subset of the data. For a discussion of typical issues involved in parallelizing stochastic gradient descent, see the recent talk by Delalleau and Bengio [12].

4.2 ϵ-accurate SVM Regression

SVM Regression is another state of the art method for regression tasks [29]. However, SVM solvers typically scale poorly with large training set sizes. We considered the use of parallelized SVMs, but preferred a faster method that yields an ϵ-accurate model: the Pegasos (Primal Estimated sub-GrAdient SOlver for SVM) algorithm [30]. This iterative SVM solver is especially well-suited for learning from large datasets. It proposes a method that alternates between two steps: stochastic sub-gradient descent and projection of the hypothesis back to the feasible set. This allows for aggressive updates that achieve convergence to an ϵ-accurate model within $O(\frac{1}{\epsilon})$ iterations, rather than the more typical $O(\frac{1}{\epsilon^2})$ requirement. Because this convergence rate does not depend on the size of the training set, the Pegasos algorithm is well suited to solving large scale SVM Regression problems.

We use Pegasos to solve the following SVM Regression optimization problem:

$$\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \sum_{\mathbf{x}\in S} log(1 + e^{\mathbf{w}\cdot\mathbf{x} - y_{\mathbf{x}} - \gamma}) + log(1 + e^{y_{\mathbf{x}} - \mathbf{w}\cdot\mathbf{x} - \gamma})$$

Here, λ is a regularization parameter, γ is a the regression shift parameter and $||\mathbf{w}||$ is the L2-norm of $\mathbf{w}$. The SVM prediction in this formulation is not guaranteed to be in [0,1], so we use a logistic transform on the SVM output to convert it to a probability estimate [25].

4.3 Feature Types

To train a model for this task, it is necessary to extract features from the content of each advertisement. These features are summarized in Table 1. We describe additional details regarding these features here.

The **parsed terms** were extracted from each content source using a multi-lingual lexicon, and were scored using methods similar to TF-IDF. The top ten scoring terms per source were considered "primary" parsed terms, and the next forty terms were considered "secondary" and placed in a distinct feature group. These were each converted to a binary score using a function $\delta(x)$ that returns 1 iff x is a term in the group under consideration, and 0 otherwise.

The **related terms** were derived from the parsed terms using a transformation $\phi(\cdot)$, using a proprietary process similar to term expansion via latent semantic analysis [11].

Cluster membership shows the strength of similarity of a given piece of content to a set of topical clusters M, as determined by a mapping function $m(\cdot,\cdot)$. These topical clusters M were found by a proprietary process similar to latent semantic analysis.

Category membership is similar to cluster membership, except that the set of categories V is drawn from a semi-automatically constructed hierarchical taxonomy.

Shannon redundancy was intended to give a sense of how "topical" a particular piece of content is. The idea here is that pages which are topically focused will have term distributions much further from uniform than less focused pages. Note that in the formula given in Table 1, $H(\mathbf{p})$ is the entropy of the distribution of landing page terms p.

Binary Cosine similarity between content groups gives a traditional measure of relevance, computed with the standard dot product and L2-norm $||\cdot||$. We use binary scoring of parsed terms, assigning weights of 1 to each parsed term appearing in the content and 0 to each non-appearing term.

Binary Kullback-Leibler (KL) divergence, like cosine similarity, was intended to provide a traditional term-based relevance measure. As before, "binary" means that we assign weights of 1 to all items in the term vector. To avoid zero probabilities, we smoothed the probability distributions P and Q (i.e. the term vectors) using Good-Turing discounting [13] before computing the divergence. We also computed a normalized version of KLD whose range is [0,1], with the maximum KLD as the normalization factor.

4.4 Evaluation

Given a feature mapping $\mathbf{x}(\mathbf{q},\mathbf{c},\mathbf{p})$ and a logistic regression function $f(\mathbf{x}) : U \mapsto [0,1]$, we evaluated its performance on predicting the bounce rate for a previously unseen subset $S \subset U$ using the three standard measures: mean absolute error, mean squared error, and correlation.

Mean Squared Error (MSE). This is given by the sum of squared deviations between the actual probability value and the predicted value, on the unseen data S:

$$L_{mse} = \frac{1}{|S|} \sum_{(\mathbf{q},\mathbf{c},\mathbf{p})\in S} \left(f(\mathbf{x}) - B_{rate}(\mathbf{q},\mathbf{c},\mathbf{p})\right)^2$$

Method	MSE	MAE	Corr.
AdCorpus1			
Logistic Regression	0.0146	0.0903	0.303
Pegasos SVM Regression	0.0148	0.0916	0.315
Baseline	0.0160	0.0959	-
AdCorpus2			
Logistic Regression	0.0148	0.0912	0.330
Pegasos SVM Regression	0.0149	0.0917	0.338
Baseline	0.0165	0.0975	-

Table 2: Results for Predicting Bounce Rate. 95% confidence intervals are on the order of ± 0.0001 for MAE and ± 0.00005 for MSE.

The ideal value for MSE is 0, with larger values showing more error.

Mean Absolute Error (MAE). This is given by the sum of absolute deviations between the actual probability value and the predicted value, on the unseen data S:

$$L_{mae} = \frac{1}{|S|} \sum_{(\mathbf{q},\mathbf{c},\mathbf{p})\in S} |f(\mathbf{x}) - B_{rate}(\mathbf{q},\mathbf{c},\mathbf{p})|$$

The ideal value for MAE is 0, with larger values showing more error. Because we are predicting probabilities in the range $[0,1]$, MAE emphasizes the impact of large errors in prediction more than MSE.

Correlation. Pearson's correlation $\rho_{f(\cdot),S}$ between the bounce rate predicted by $f(\cdot)$ and true bounce rate across all examples in the unseen data S is given by:

$$\rho_{f(\cdot),S} = \frac{\sum_{(\mathbf{q},\mathbf{c},\mathbf{p})\in S} f(\mathbf{x}) B_{rate}(\mathbf{q},\mathbf{c},\mathbf{p}) - |S|\mu_{f(\cdot)}\mu_{B_{rate}(\cdot)}}{(|S|-1)\sigma_{f(\cdot)}\sigma_{B_{rate}(\cdot)}}$$

where σ is standard deviation and μ is the observed sample mean. This correlation coefficient is helpful for detecting the presence of informative predictions, even in the presence of shifting and scaling. The ideal value for correlation is 1.0, with a value of 0 showing no observed correlation.

5. EXPERIMENTAL RESULTS

In this section, we report experimental results showing that it is, indeed, possible to predict bounce rate using features from the advertisement content alone. We show that these predictions are stable over time, and analyze the impact of specific feature types.

5.1 Data sets

We created two large datasets AdCorpus1 and AdCorpus2, consisting of ads that got at least one click in AdWords in two disjoint time periods in 2008, where the time period of AdCorpus2 was after the time period corresponding to AdCorpus1. Each data set was split randomly into training and test sets, using standard machine learning evaluation methodology, with 70% training and 30% test. Each example had at least 10 observed clicks, to allow for reasonable evaluation of true bounce rate in the final evaluation, although most examples has significantly more. The training and test sets for AdCorpus1 had 3.5 million and 1.5 million data points respectively, while the training and test sets for AdCorpus2 had 4.8 million and 2 million data points respectively.

Feature Type	# Unique Features	Values	Formula	Notes
Parsed Keyword Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{q})$	Top 10 selected
Parsed Creative Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{c})$	Top 10 selected
Parsed Landing Page Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{p})$	Top 10 selected
Parsed Keyword Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{q})$	Top 11-50 selected
Parsed Creative Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{c})$	Top 11-50 selected
Parsed Landing Page Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \mathbf{p})$	Top 11-50 selected
Related Keyword Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{q}))$	Top 10 selected
Related Creative Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{c}))$	Top 10 selected
Related Landing Page Terms, Primary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{p}))$	Top 10 selected
Related Keyword Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{q}))$	Top 11-50 selected
Related Creative Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{c}))$	Top 11-50 selected
Related Landing Page Terms, Secondary	Millions	$\{0,1\}$	$\delta(x \in \phi(\mathbf{p}))$	Top 11-50 selected
Keyword Cluster Membership, Primary	Thousands	$\mathbb{R}$	$m(\mathbf{q}, M)$	Top 4 selected
Creative Cluster Membership, Primary	Thousands	$\mathbb{R}$	$m(\mathbf{c}, M)$	Top 4 selected
Landing Page Cluster Membership, Primary	Thousands	$\mathbb{R}$	$m(\mathbf{p}, M)$	Top 4 selected
Keyword Cluster Membership, Secondary	Thousands	$\mathbb{R}$	$m(\mathbf{q}, M)$	Top 5-10 selected
Creative Cluster Membership, Secondary	Thousands	$\mathbb{R}$	$m(\mathbf{c}, M)$	Top 5-10 selected
Landing Page Cluster Membership, Secondary	Thousands	$\mathbb{R}$	$m(\mathbf{p}, M)$	Top 5-10 selected
Keyword Categories, Primary	Hundreds	$\mathbb{R}$	$m(\mathbf{q}, V)$	Top 2 selected
Creative Categories, Primary	Hundreds	$\mathbb{R}$	$m(\mathbf{c}, V)$	Top 2 selected
Landing Page Categories, Primary	Hundreds	$\mathbb{R}$	$m(\mathbf{p}, V)$	Top 2 selected
Keyword Categories, Secondary	Hundreds	$\mathbb{R}$	$m(\mathbf{q}, V)$	Top 3-10 selected
Creative Categories, Secondary	Hundreds	$\mathbb{R}$	$m(\mathbf{c}, V)$	Top 3-10 selected
Landing Page Categories, Secondary	Hundreds	$\mathbb{R}$	$m(\mathbf{p}, V)$	Top 3-10 selected
Shannon Redundancy	1	$\mathbb{R}$	$1 - \frac{H(\mathbf{p})}{log\|p\|}$	Distance of landing page term distribution from uniform
Keyword to Creative Cosine Sim.	1	[0,1]	$\frac{\mathbf{q}\cdot\mathbf{c}}{\|\mathbf{q}\|\|\mathbf{c}\|}$	Binary term weighting
Keyword to Landing Page Cosine Sim.	1	[0,1]	$\frac{\mathbf{q}\cdot\mathbf{p}}{\|\mathbf{q}\|\|\mathbf{p}\|}$	Binary term weighting
Creative to Landing Page Cosine Sim.	1	[0,1]	$\frac{\mathbf{c}\cdot\mathbf{p}}{\|\mathbf{c}\|\|\mathbf{p}\|}$	Binary term weighting
Keyword to Creative KLD	1	$\mathbb{R}$	$\sum_{x\in\Sigma} C(x) \log \frac{C(x)}{Q(x)}$	Good-Turing term smoothing
Keyword to Landing Page KLD	1	$\mathbb{R}$	$\sum_{x\in\Sigma} P(x) \log \frac{P(x)}{Q(x)}$	Good-Turing term smoothing
Creative to Landing Page KLD	1	$\mathbb{R}$	$\sum_{x\in\Sigma} P(x) \log \frac{P(x)}{C(x)}$	Good-Turing term smoothing

Table 1: Details of features used for training the bounce rate prediction models.

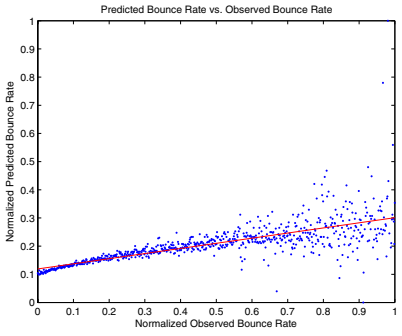

Figure 5: Scatter plot of binned predicted bounce rates versus observed (true) bounce rates for logistic regression model using all features, on AdCorpus1.

5.2 Primary Results

For our primary experiment, we tested our parallelized logistic regression method and Pegasos SVM regression method against a baseline method of predicting the mean value for all examples using the train/test splits in AdCorpus1 and AdCorpus2. We used all features described in Section 4.3.

For logistic regression, the λ parameter was set to 0.2. For Pegasos SVM regression, λ was set to 0.001. These values were selected as reasonable default settings, and were not tuned to the data. (Pilot experiments on similar previous data sets suggested that results were similar across a range of λ values for logistic regression.)

The parallelized logistic regression was trained on the full training set, using multiple machines. For the Pegasos algorithm for SVM regression, we used a sample of 250,000 examples randomly selected from the training set so that the data would fit into 8G of RAM on a single machine, and ran the algorithm for 25 million iterations. Pegasos training finished within an hour on the machine, which had 2 cores (2.2 GHz each).

The results, given in Table 2, show a clear win for the machine learning approaches over the baseline approach. The relative improvements over the baseline methods range from 5% to 10% reduction in MAE and MSE. The scatter plot of predicted bounce rate against true bounce rate, shown in Figure 5 shows that the predictions are typically more accurate at the lower (more common) end of the spectrum. While

Train	Test	MSE	MAE	Corr.
AdCorpus1	AdCorpus1	0.0146	0.0903	0.303
AdCorpus1	AdCorpus2	0.0150	0.0912	0.310
AdCorpus2	AdCorpus2	0.0148	0.0912	0.330
AdCoprus2	AdCorpus1	0.0146	0.0906	0.338

Table 3: Results for Predicting Bounce Rates Across Time.

Feature Set	MSE	MAE	Corr.
All	0.0146	0.0903	0.303
Parsed (Kw+Cr+Lp)	0.0151	0.0921	0.248
Parsed Creatives	0.0153	0.0926	0.222
Related (Kw+Cr+Lp)	0.0153	0.0927	0.227
Categories (Kw+Cr+Lp)	0.0154	0.0929	0.208
Parsed Landing Page	0.0155	0.0932	0.198
Clusters (Kw+Cr+Lp)	0.0155	0.0934	0.197
Cos. + ShanRed + KLD	0.0155	0.0941	0.180
Parsed Keywords	0.0156	0.0942	0.154

Table 4: Results for Predicting Bounce Rate for Restricted Feature Sets on AdCorpus1 using Logistic Regression. 95% confidence intervals are on the order of ± 0.0001 for MAE and ± 0.00005 for MSE.

the gains made by the machine learning methods may appear at first as a small improvement, paired t-tests measured the improvements as significant with $p < 10^{-6}$. Furthermore, if these predictions can be used to improve advertisement conversion rates by similar figures, then the advertisers will reap a significant increase in return on investment.

These results demonstrate that it is possible to learn to predict bounce rate from content of the advertisement alone. Furthermore, we observe that Pegasos SVM regression, trained on a single machine on a subset of the data, gave results quite close to the parallelized logistic regression method using the full training data across several machines.

5.3 Stability Over Time

Do these models generalize well into the future, or is their predictive capability limited to a small time-frame? We addressed this question by running a stability experiment. We trained a logistic regression model on the AdCorpus1 training data, and tested this model on both the AdCorpus1 test data and the later AdCorpus2 test data. (We did not separately test the Pegasos SVM regression, as this had given nearly identical results to logistic regression in the primary experiment.)

The results, shown in Table 3, show that the model trained on AdCorpus1 did *not* lose its effectiveness on later data given in AdCorpus2. The MSE, MAE, and Correlation values are all quite close. We repeated this experiment in reverse, training on AdCorpus2 training data, and testing on AdCorpus2 and AdCorpus1 data sets with similar results, also shown in Table 3. This is evidence that the models are learning fundamental qualities in advertisements that do not change dramatically over time.

5.4 Analysis of Features

We have seen that using all of the features described in Section 4.3 allows bounce rates to be learned effectively.

	Kywd.	Crtv.	LandPg.	Dists.
Keyword	1	0.339	0.300	0.103
Creative	-	1	0.435	0.183
Landing Page	-	-	1	0.146

Table 5: Correlation between predictions from models with different feature sources: keywords only, creatives only, landing pages only, and features using only the group of cosine similarity, KLD, and Shannon Redundancy.

Here, we examine the impact of specific feature types in bounce rate prediction.

We first trained logistic regression models using particular subsets of all available features, and measured the performance of the models using each feature subset. The results of this are given in Table 4. Not surprisingly, the model using all features in combination gave best results. However, the relative performance of feature sub-sets is informative. Table 4 shows that the parsed term features drawn from keywords, creatives, and landing pages gives the best performance of any of the subsets. That is, specific terms influence bounce rates more strongly than general topics.

Additionally, we observe the counter-intuitive result that terms from the creative text are more informative than those drawn from landing pages or keywords. This agrees with observations from Section 3.2, connecting bounce rate and CTR. This suggests that advertisers may be able to improve their bounce rates not only by improving the quality of their landing page, but also by improving the quality of their creative text. Finally, note that parsed terms from keywords are the least informative features, despite the wide variance in bounce rates by keywords shown in Section 3.5. This suggests that bounce rates are dependent on more than simply choosing "quality" keywords – they depend equally on the qualities of the advertisement itself.

We continue this investigation by examining the correlation coefficients computed on the predictions from these distinct models. Table 5 shows the correlation coefficients between predictions from models trained on all keyword features, all creative features, all landing page features, and the "distance" features of cosine similarity, Shannon Redundancy, and KLD. We can see that the models using creative features only and landing page features only are highly correlated, supporting the observation that creative texts capture (or communicate to users) much of the information of the landing page. Interestingly, the "distance" features give predictions relatively uncorrelated with the other feature sources. Similar results are seen in Table 6, which group features by type rather than by source. This finding may enable the use of semi-supervised learning methods such as co-training that require informative but un-correlated feature sets [4] to exploit un-labeled data.

6. RELATED WORK

To our knowledge, this work is the first detailed study of bounce rate for sponsored search advertising. It also provides the first concrete proposal for predicting bounce rates in the absence of historical clickthrough data. This task is related to prior work in predicting textual relevance and in modeling user behavior, as reviewed in the remainder of this section.

	Parsed	Related	Clusters	Categories	Distances.
Parsed	1	0.492	0.444	0.397	0.188
Related	-	1	0.534	0.499	0.161
Clusters	-	-	1	0.412	0.127
Categories	-	-	-	1	0.180

Table 6: Correlation between predictions from models with different feature types.

6.1 Textual Relevance

Among other features, our models for predicting bounce rates use measures of textual relevance. Such measures have been studied in the context of the impedance mismatch problem in contextual advertising, which refers to the mismatch between the vocabularies of publisher pages and textual advertisements. Researchers in computational advertising have suggested various methods to address this issue in order to design good matching functions between publisher pages and ads.

Broder et al. [7] found that while training a model to predict the relevance of an ad to a publisher page, it is useful to augment "syntactic" features obtained by matching keywords or phrases between the ad and the page with "semantic features" obtained by categorizing ads and pages into a commercial taxonomy to calculate their topic similarity. Their experiments showed that a convex linear combination of syntactic and semantic features had an improvement over syntactic features alone, with respect to a "golden ranking" produced by human relevance judgements.

Murdock et al. [22] showed the benefit of using machine translation techniques to match text features extracted from ads to those obtained from publisher pages in order to address the impedance problem. They obtained better results by adding text features from the landing pages of ads, and improved the ranking of content ads shown on a publisher page by using a support vector machine (SVM) based ranking model.

Riberio-Neto et al. [27] proposed a Bayesian network-based approach to impedance coupling, for better matching of ads to publisher pages in contextual advertising. They also proposed different strategies for improving relevance-based matching functions.

Chakrabarti et al. [9] used clicks on contextual ads to learn a matching function. They trained a logistic model for predicting ad clicks based on relevance features between the publisher page and the ad. By training the features of the logistic model using click logs, they outperformed traditional page-ad matching functions that use hand-tuned combinations of syntactic or semantic features from the ad or page text.

Textual relevance has also been used for other problems in sponsored search. Broder et al. [6] have used terms from the search results to enhance query terms for selecting advertisements. They demonstrated that the careful addition of terms from the web search results (extracting relevant phrases from search results, using both keyword-level and topic-level features) to the query terms can improve the retrieval of relevant ads. Their approach performed better when compared to a system that augments the query terms by phrases extracted from web users' query rewrites in search logs. Radlinski et al. [26] also showed that relevance between query and ad text can improve broad match while optimizing revenue.

Other notable uses of relevance in computational advertising includes learning when not to show ads. Broder et al. [5] trained an SVM model using relevance and cohesiveness features to address the decision problem of whether or not to show an ad.

6.2 Modeling User Behavior

Another aspect of our work is modeling user behavior on ad landing pages. While bounce rate has not been modeled by researchers in the past, other aspects of user clickthrough behavior have been studied in the context of evaluating the quality of both ad and search results.

Ciaramita et al. [10] estimated predicted clickthrough rates of search ads from textual relevance features. They trained a logistic model whose features were learned using clickthrough data from logs. Their work demonstrated that simple syntactic and semantic textual relevance features can be predictive of clickthrough rate.

Piwowarski et al. [24] modeled user clickthrough on search results using specific click history (e.g., from users) or more general click history features (e.g., from user communities or global history).

Agichtein et al. [1] used click behavior to improve search ranking. They observed that while individual clicks may be noisy, aggregating clicks to get overall statistics (e.g., clickthrough rate) gives reliable estimates that can be used to re-rank search results for queries and get quality improvements. Agichtein et al. [2] also developed a model for relating user behavior to relevance, proposing a simple linear mixture model for relating observed post-search user behavior to the relevance of a search result.

Huffman et al. [14] examined the connection between search-result relevance in web search and users' session-level satisfaction. They found a strong relationship between the relevance of the first query in a user session and the user's satisfaction in that session, and built a model that predicts user satisfaction by incorporating features from the user's first query into a relevance model. Their models were evaluated using relevance judgements of human raters.

Carterette et al. [8] have used click information to evaluate the performance of search results – they proposed a model for predicting relevance of a search result to a user using clickthrough information. Such a model would be useful for evaluating search results for which human relevance judgments have not been obtained yet.

Pandey et al. [23] proposed a multi-arm bandit approach with dependent arms for more accurate clickthrough prediction, using historical observation along with other features such as textual similarity between ads.

7. CONCLUSIONS AND FUTURE WORK

This paper has demonstrated through quantitative and qualitative analysis that bounce rate provides a useful assessment of user satisfaction for sponsored search advertising

that complements other quality metrics such as clickthrough and conversion rates. We described methods of estimating bounce rate through observing user behavior, and have provided extensive analysis of real world bounce rate data to develop the reader's understanding of this important metric. We have also shown that even in absence of substantial clickthrough data, bounce rate may be estimated through machine learning when applied to features extracted from sponsored search advertisements and their landing pages. These improvements in predictions over baseline methods were statistically significant, and would be sufficient to drive solid gains in advertiser return on investment assuming they translate into improved conversion rates. We continue to pursue additional improvements in estimation accuracy, and believe one promising avenue for improvement is the use of link-based or other non-textual features.

In closing, we note that while bounce rate can be useful for identifying ad quality problems, bounce rate alone does not immediately suggest what actions an advertiser can take to address them. In future work, we intend to study how advertisers might best act on bounce rate information in order to improve their advertising return on investment.

Acknowledgments

We would like to thank Ashish Agarwal, Vinay Chaudhary, Deirde O'Brien, Daryl Pregibon, Yifan Shi, and Diane Tang for their contributions to this work. We also thank Josh Herbach, Andrew Moore, and Sridhar Ramaswamy for their valuable feedback.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, 2006.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR*, 2006.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[5] A. Broder, M. Ciaramita, M. Fontoura, E. Gabrilovich, V. Josifovski, D. Metzler, V. Murdock, and V. Plachouras. To swing or not to swing: Learning when(not) to advertise. In *CIKM*, 2008.

[6] A. Broder, P. Ciccolo, M. Fontoura, E. Gabrilovich, V. Josifovski, and L. Riedel. Search advertising using web relevance feedback. In *CIKM*, 2008.

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

[8] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *NIPS*, 2007.

[9] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW*, 2008.

[10] M. Ciaramita, V. Murdock, and V. Plachouras. Online learning from click data for sponsored search. In *WWW*, 2008.

[11] S. Deerwester, S. Dumais, T. Landuaer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 1990.

[12] O. Delalleau and Y. Bengio. Parallel stochastic gradient descent. In *CIAR Summer School, Toronto*, 2007.

[13] W. A. Gale and G. Sampson. Good-turing frequency estimation without tears. *J. of Quantitative Linguistics*, 2:217–237, 1995.

[14] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *SIGIR*, 2007.

[15] T. Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.

[16] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*, 2005.

[17] A. Kaushik. Bounce rate as sexiest web metric ever. MarketingProfs, August 2007. http://www.marketingprofs.com/7/bounce-rate-sexiest-web-metric-ever-kaushik.asp?sp=1.

[18] A. Kaushik. Excellent analytics tip 11: Measure effectiveness of your web pages. Occam's Razor (blog), May 2007. http://www.kaushik.net/avinash/2007/05/excellent-analytics-tip-11-measure-effectiveness-of-your-web-pages.html.

[19] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, 1997.

[20] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. In *NIPS*, 2008.

[21] D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528, 1989.

[22] V. Murdock, M. Ciaramita, and V. Plachouras. A noisy-channel approach to contextual advertising. In *AD-KDD Workshop in KDD*, 2007.

[23] S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. In *ICML*, 2007.

[24] B. Piwowarski and H. Zaragoza. Predictive user click models based on click-through history. In *CIKM*, 2007.

[25] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood models. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[26] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. Optimizing relevenace and revenue in ad search: A query substitution approach. In *SIGIR*, 2008.

[27] B. Ribeiro-Neto, M. Cristo, P. B. Golgher, and E. S. de Moura. Impedance coupling in content-targeted advertising. In *SIGIR*, 2005.

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

[29] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[30] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for Svm. In *ICML*, 2007.