MLOAD
2010

NIPS 2010 Workshop

Machine Learning in Online Advertising

held in conjunction with the 24th Annual Conference on

Neural Information Processing Systems

10 Dec. 2010, Whistler

Organizers

Deepak K. Agarwal (Yahoo! Research)

Tie-Yan Liu (Microsoft Research Asia)

Tao Qin (Microsoft Research Asia)

James G. Shanahan (Independent Consultant)

http://research.microsoft.com/~mload-2010

Preface

Online advertising, a form of advertising that utilizes the Internet and World Wide Web to deliver marketing messages and attract customers, has seen exponential growth since its inception over 15 years ago, resulting in a $65 billion market worldwide in 2008; it has been pivotal to the success of the World Wide Web.

The dramatic growth of online advertising poses great challenges to the machine learning research community and calls for new technologies to be developed. Online advertising is a complex problem, especially from machine learning point of view. It contains multiple parties (i.e., advertisers, users, publishers, and ad platforms such as ad exchanges), which interact with each other harmoniously but exhibit a conflict of interest when it comes to risk and revenue objectives. It is highly dynamic in terms of the rapid change of user information needs, non-stationary bids of advertisers, and the frequent modifications of ads campaigns. It is very large scale, with billions of keywords, tens of millions of ads, billions of users, millions of advertisers where events such as clicks and actions can be extremely rare. In addition, the field lies at intersection of machine learning, economics, optimization, distributed systems and information science all very advanced and complex fields in their own right. For such a complex problem, conventional machine learning technologies and evaluation methodologies are not be sufficient, and the development of new algorithms and theories is sorely needed.

The goal of this workshop is to overview the state of the art in online advertising, and to discuss future directions and challenges in research and development, from a machine learning point of view. We expect the workshop to help develop a community of researchers who are interested in this area, and yield future collaboration and exchanges.

Our call for papers has attracted many submissions. All submitted papers were thoroughly reviewed by the program committee. The program committee finally accepted 7 papers.

We are grateful to the program committee members for carefully reviewing all the submissions. We also would like to thank the MLOAD 2010 program committee for their support of this workshop and all the authors for their contributions.

Deepak Agarwal
Tie-Yan Liu
Tao Qin
James Shanahan
Program Committee Co-Chairs

MLOAD 2010 Organization

Program Co-Chair:

Deepak K. Agarwal (Yahoo! Research)
Tie-Yan Liu (Microsoft Research Asia)
Tao Qin (Microsoft Research Asia)
James G. Shanahan (Independent Consultant)

Program Committee:

Misha Bilenko (Microsoft)
Olivier Chapelle (Yahoo! Research)
Jon Feldman (Google)
Bin Gao (Microsoft Research Asia)
Thore Graepel (Microsoft Research Cambridge)
Diane Lambert (Google)
S. Muthukrishnan (Google)
Kishore Papineni (Yahoo! Research)
Dou Shen (Microsoft)
Dirk Van den Poel (University Gent)
Jun Wang (University College London)
Hwanjo Yu (POSTECH)

MLOAD 2010 Schedule

7:30-7:45	**Opening Remarks**
7:45-8:30	**Keynote: Machine Learning for Display Advertising**
	Foster Provost
8:30-9:00	**Invited talk: AdPredictor - Large Scale Bayesian Click-Through Rate Prediction in Microsoft's Bing Search Engine**
	Thore Graepel, Joaquin Quinonero Candela
9:00-9:30	Cofffeee Break
9:30-10:00	**Invited talk: Hybrid Bidding for Keyword Auctions**
	Ashish Goel
10:00-10:30	**Poster Boaster and Discussion Sessions**
10:30-15:30	Ski Break
15:30-16:15	**Keynote: Visualization and Modeling of the Joint Behavior of Two Long Tailed Random Variables**
	Art Owen
16:15-16:45	**Invited talk: Click Modeling in Search Advertising: Challenges and Solutions**
	Jianchang Mao
16:45-17:05	**Invited talk: Digital Advertising: Going from Broadcast to Personalized Advertising**
	James Shanahan
17:05-17:30	Coffee Break
17:30-17:50	**Invited talk: Machine Learning for Advertiser Engagement**
	Tao Qin
17:50-18:30	**Panel Discussions**

Table of Contents

Keynote Addresses

Invited Talks

Accepted Papers

Keynote address

Machine Learning for Display Advertising

Foster Provost
New York University

Abstract

Most on-line advertisements are display ads, yet as compared to sponsored search, display advertising has received relatively little attention in the research literature. Nonetheless, display advertising is a hotbed of application for machine learning technologies. In this talk, I will discuss some of the relevant differences between online display advertising and traditional advertising, such as the ability to profile and target individuals and the associated privacy concerns, as well as differences from search advertising, such as the relative irrelevance of clicks on ads and the concerns over the content next to which brands' ads appear. Then I will dig down and discuss how these issues can be addressed with machine learning. I will focus on two main results based on work with the successful machine-learning based firm Media6degrees. (i) Privacy-friendly ``social targeting'' can be quite effective, based on identifying browsers that share fine-grained interests with a brand's existing customers--as exhibited through their browsing behavior. (ii) Clicks often are a poor surrogate for conversions for training targeting models, but there are effective alternatives.

This work was done in collaboration with Brian Dalessandro, Rod Hook, Alan Murray, Claudia Perlich, and Xiaohan Zhang.

Keynote address

Visualization and Modeling of the Joint Behavior of Two Long Tailed Random Variables

Art Owen
Stanford University

Abstract

Many of the variables relevant to online advertising have heavy tails. Keywords range from very frequent to obscure. Advertisers span a great size range. Host web sites range from very popular to rarely visited.

Much is known about the statistical properties of heavy tailed random variables. The Zipf distribution and Zipf-Mandelbrot distribution are frequently good approximations.

Much less attention has been paid to the joint distribution of two or more such quantities. In this work, we present a graphical display that shows the joint behavior of two long tailed random variables. For ratings data (Netflix movies, Yahoo songs) we often see a strong head to tail affinity where the major players of one type are over-represented with the minor players of the other. We look at several examples which reveal properties of the mechanism underlying the data. Then we present some mathematical models based on bipartite preferential attachment mechanisms and a Zipf-Poisson ensemble.

This is joint work with Justin Dyer.

Invited Talk

Hybrid Bidding for Keyword Auctions

Ashish Goel
Stanford University

Abstract

Search auctions have become a dominant source of revenue generation on the Internet. Such auctions have typically used per-click bidding and pricing. We propose the use of hybrid auctions where an advertiser can make a per-impression as well as a per-click bid, and the auctioneer then chooses one of the two as the pricing mechanism. We assume that the advertiser and the auctioneer both have separate beliefs (called priors) on the click-probability of an advertisement. We first prove that the hybrid auction is truthful, assuming that the advertisers are risk-neutral. We then show that this auction is different from the existing per-click auction in multiple ways: 1) It takes into account the risk characteristics of the advertisers. 2) For obscure keywords, the auctioneer is unlikely to have a very sharp prior on the click- probabilities. In such situations, the hybrid auction can result in significantly higher revenue. 3) An advertiser who believes that its click-probability is much higher than the auctioneer's estimate can use per-impression bids to correct the auctioneer's prior without incurring any extra cost. 4) The hybrid auction can allow the advertiser and auctioneer to implement complex dynamic programming strategies. As Internet commerce matures, we need more sophisticated pricing models to exploit all the information held by each of the participants. We believe that hybrid auctions could be an important step in this direction.

Invited Talk

AdPredictor – Large Scale Bayesian Click-Through Rate Prediction in Microsoft's Bing Search Engine

Thore Graepel and Joaquin Quiñonero Candela
Microsoft

Abstract

In the past years online advertising has grown at least an order of magnitude faster than advertising on all other media. Bing and Yahoo! have recently joined forces: all ads on both search engines are now served by Microsoft adCenter and all search results on Yahoo! are powered by Bing. Accurate predictions of the probability that a user clicks on an advertisement for a given query increase the efficiency of the ads network and benefit all three parties involved: the user, the advertiser, and the search engine. This talk presents the core machine learning model used by Microsoft adCenter for click prediction: an online Bayesian probabilistic classification model that has the ability to learn efficiently from terabytes of web usage data. The model explicitly represents uncertainty allowing for fully probabilistic predictions: 2 positives out of 10 instances or 200 out of 1000 both give an average of 20%, but in the first case the uncertainty about the prediction is larger. We discuss some challenges in machine learning for online systems, such as valid metrics, causal loops and biases in the training data.

Invited Talk

Click Modeling in Search Advertising: Challenges and Solutions

Jianchang Mao
Yahoo! Labs

Abstract

Sponsored search is an important form of online advertising that serves ads that match user's query on search result page. The goal is to select an optimal placement of eligible ads to maximize a total utility function that captures the expected revenue, user experience and advertiser return on investment. Most search engines use a pay-per-click model where advertisers pay the search engine a cost determined by an auction mechanism (e.g., generalized second price) only when users click on their ad. In this case, the expected revenue is directly tied to the probability of click on ads. Click is also often used as a proxy for measuring search user experience, and is a traffic driver for advertisers. Therefore, estimation of the probability of click is the central problem in sponsored search. It affects ranking, placement, quality filtering and price of ads.

Estimating click probability given a query-ad-user tuple is a challenging statistical modeling problem for a large variety of reasons, including click sparsity for the long tail of query-ad-user tuples, noisy clicks, missing data, dynamic and seasonal effects, strong position bias, selection bias, and externalities (context of an ad being displayed). In this talk, I will provide an overview on some of the machine learning techniques recently developed in Advertising Sciences team at Yahoo! Labs to deal with those challenges in click modeling. In specific, I will briefly describe: (i) a temporal click model for estimating positional bias, externalities, and unbiased user-perceived ad quality in a combined model; (ii) techniques for reducing sparsity by aggregating click history for sub-queries extracted with a CRF model and by leveraging data hierarchies; and (iii) use of a generative model for handling missing click history features. The talk is intended to give a flavor of how machine learning techniques can help solve some of the challenging click modeling problems arising in online advertising.

Invited Talk

Digital Advertising: Going from Broadcast to Personalized Advertising

James G. Shanahan
Independent Consultant

Abstract

Online advertising is a form of promotion that uses the Internet and World Wide Web for the expressed purpose of delivering marketing messages to attract customers. Examples of online advertising include text ads that appear on search engine results pages, banner ads, in-text ads, or Rich Media ads that appear on regular web pages, portals or applications. Since it inception over 15 years ago, online advertising has grown rapidly and currently accounts for 10\% of the overall advertising spend (which is approximately $600 billion worldwide)). A large part of the more recent success in this field has come from the following key factors:

* Personalization: offline advertising (via broadcast TV, radio, newspaper etc.) is largely a broadcast form of communication where as digital advertising is much more targeted and thus enables a personalized, and possibly informative, message to consumers.

* Interactivity: internet advertising is becoming increasingly interactive with the advent of new forms of advertising such as social advertising; this is enables advertisers and consumers to operate in a more conversant manner.

* Engagement: consumers are spending more time online than with any other form of media thereby enabling a broader reach and deeper connection with consumers.

* Explainabilty: advertisers are beginning to understand their consumers better.

This shift in focus in digital advertising from location (i.e., publisher web pages) to personalization has brought with it numerous challenges some of which have received a lot of research attention in the data mining and machine learning communities over the past 10-20 years. In this talk I will review, along the dimensions outlined above, some of these key technical problems and challenges that arise when adverting becomes personal. This will be done within the context of the elaborate (and ever-evolving) ecosystems of modern day digital advertising where one has to capture, store, and process petabytes of data within the constraints of a, sometimes, sequential workflow. The ultimate goal to is provide millisecond-based decision-making at each step of this workflow that enables customizable and engaging consumer experiences.

Invited Talk

Machine Learning for Advertiser Engagement

Tao Qin
Microsoft Research Asia

Abstract

Advertiser engagement, which goal is to attract more advertisers, make them loyal to the ad platform, and make them willing to spend more money on (online) advertising, is very important for an ad platform to boost its long-term revenue. Industry has paid more and more attention to advertiser engagement. For example, many search engines have provided tools to help advertisers, including keyword suggestion, traffic (number of impressions/clicks) estimation, and bid suggestion. However, from the research point of view, the effort on advertiser engagement is still limited.

In this talk, we discuss the challenges in advertiser engagement, especially from the machine learning perspective. Actually machine learning algorithms can be used in many aspects of online advertising, such as CTR prediction. We propose a number of principles that should be considered when using machine learning technologies to help advertiser engagement.

(1) Accurate. The results of learning algorithms should be as accurate as possible. This principle is the same as that in other machine learning tasks.

(2) Socially fair. The learning algorithms should promote diversity and be fair to even tail advertisers. In this way, more advertisers will feel engaged and the entire ads eco-system will become more healthy.

(3) Understandable. The evaluation metrics and learned models should be easy to interpret. In this way, it is easier for advertisers to diagnose their campaigns and identify the key aspects to improve. This will also make the ad platform more transparent to advertisers and increase their trust in the ad platform.

(4) Actionable. The learning algorithms should provide actionable suggestions/feedback to advertisers. In this way, the advertisers can take effective actions to improve their performances, and therefore stick to the ad platform in a more loyal fashion.

We will show several example problems in online advertising (such as effectiveness evaluation and auction mechanism) and discuss possible solutions based the above principles.

This is joint work with Bin Gao and Tie-Yan Liu.

CTR prediction based on click statistic

Bauman K.
Yandex[*]
kbauman@yandex-team.ru

Kornetova A.
Yandex
akornet@yandex-team.ru

Topinskiy V.
Yandex
vtopin@yandex-team.ru

Leshiner D.
Yandex
leshch@yandex-team.ru

Abstract

A new click through rate predicting formula which is based on statistics of clicks and shows is described. It is a result of increasing of log-likelihood given a constraint on formula's variance at small banner history. Form of this predicting formula is derived in different approaches. High risk problem of formula at small banners history is discussed.

1 Introduction

Sponsored search or Search Advertising is one of the major part of search engines' revenues. Here we consider the advertising model that concerns presenting contextual ads directly on query results pages (SERP). Advertisers are typically charged each time their ads are clicked (the pay-per-click approach).

The number of ads that the search engine can show to a user is limited and different positions on the SERP have different attractiveness: an ad shown at the top of the page is more likely to be clicked than an ad shown at the bottom. That's why search engines need a system for allocating the positions to ads and ranking by CPM (cost per million) is a natural choice.

When user asks search query the system finds all ads that are candidates to be shown. After calculating ad's CTR predictions system ranks them by $CPM = bid * CTR$ and displays best eight. Choice of CPM as ranking factor is justified by the fact that it estimates an expected payment for a banner.

The task of CTR prediction is crucial to Sponsored Search advertising because it impacts user experience, profitability of advertising and search engine revenue.

The main problem that we discuss in our paper is the part of click-through rate prediction (CTR) used for Sponsored Search.

2 Offline evaluation

To compare two CTR predictions we need to work with measures. There are some well-known measures such as log-likelihood and mean square error (MSE).

But in fact CTR prediction is only the small part of an advertising system. What is really interesting to see is how system's CTR and revenue will change if we try to use different CTR predictions.

[*] 119021, Leo Tolstoy 16, Moscow, Russia

The best but not the easiest way to do it is to make an on-line experiment but it always take a lot of time. So we developed two offline metrics.

1. Method of ROC-like curves

   Lets take a set of events from a log as input. To predict CTR we use our predictor, then we sort all events by CPM and draw following plot: CPM as X, and as Y we take a number of events from the log, which has CPM not less than corresponding value on X.

   For clicks we draw following plot: CPM as X, and as Y we take a number of clicks by events from the log, which has CPM not less than corresponding value on X. The same plot we draw for money and for CTR.

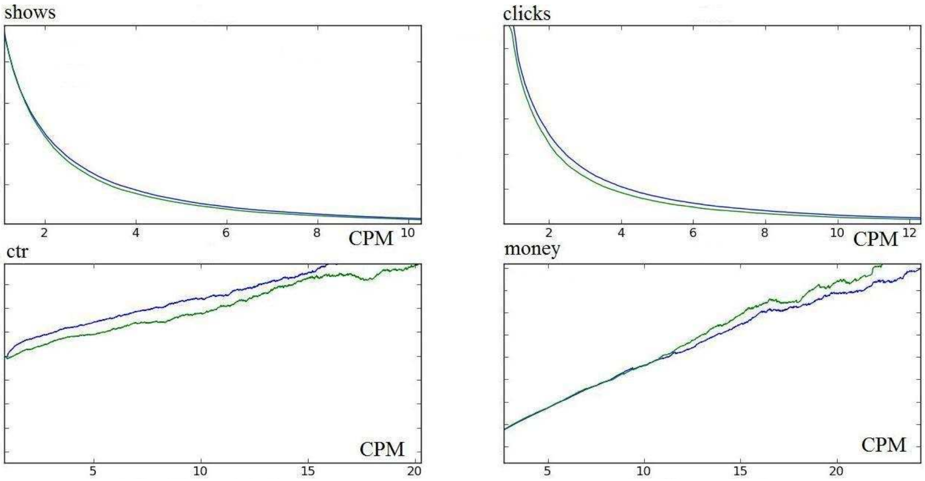


Figure 1: Method of ROC-like curves

   This plot can help us to compare different predictors. If curve of the first predictor is always over the other, so it means that if we through out some number of the worst shows from log, the first predictor will save better events then other. It will have more money and more clicks.

   This method has one deficiency. Curves of two predictors can be different only in interior area of a CPM range. But if we take the ends of this interval we will see that all curves has the same value.

2. Model of system

   We get a copy of whole our online advertising system. Then we bind this offline copy with the current snapshots of all needed databases.

   Now we can implement there a prediction of CTR and run this specified system on some pool of queries. Output of this execution for one query is a pool of all relevant ads.

   Then we get two sets of results for total queries' pool for our predictions respectively, calculate expected revenue and average CTR for further comparison. We use better in log-likelihood CTR prediction to calculate interested us expectations.

   So we get a few numeric properties of our predictions and compare them by means of all this features.

3 Inference of optimal form of the formula

3.1 Baseline formula

Every banner whenever shown in our system has its history of clicks and shows. Hereafter $clicks_b$ is a number of observed banner b's clicks in sample of this banner's shows and $shows_b$ is a number of observations in this sample. Let A is a set of advertisement banners for which we collect statistics $clicks$ and $shows$. We assume that click on a banner is generated like random variate with respect to Bernoulli distribution with $p = ctr_b$. It's known that for one banner the maximum likelihood estimation of ctr is the ratio of clicks to shows. But it's also known that this estimation has good performance only when we have a sufficiently large number of observations.

As a baseline we have some formula that corresponds to a constant for a banner with no history and then it somehow converges to the ratio of clicks to shows. This formula predicts CTR using only statistics of shows and clicks of banner.

Our goal is to make the best function with the same properties.

3.2 Sketch of the inference in MSE framework

We consider question about how we should take into account statistics of clicks with respect of amounts of observations to make some predictions about click probabilities on advertisement banners better in some sense than the sample mean.

Suppose that we have some additional information about C set of $\{ctr_b | b \in A\}$ and this information is $\bar{ctr}_b$ average of ctr_b from C. Then we wish to do some trade-off between the maximum likelihood estimation and this average with respect to a number of observations. More formally we try to find a ctr prediction as linear combination, coefficients of which are functions of shows.

$$\hat{ctr}(shows) = \alpha(shows) * \bar{ctr} + \beta(shows) * \frac{clicks}{shows},\ where\ \bar{ctr} = mean_{b \in A}(ctr_b)$$

One way to find unknown coefficients is to choose them by means of minimizing of expected mean square errors $Q(shows)$.

$$Q(shows) = E \sum_{b \in A} (ctr_b - \hat{ctr}_b(shows))^2.$$

Then it's easy to show that optimal coefficients are calculated as follows.

$$\alpha(shows) + \beta(shows) = 1,\ \alpha(shows) = \frac{S_0}{S_0 + shows},$$

$$where\ S_0 = \frac{mean_{b \in A}(ctr_b * (1 - ctr_b))}{var_{b \in A}(ctr_b)}$$

Now we can rewrite our new ctr predictions as follows.

$$\hat{ctr} = \alpha(clicks) + \beta(clicks) * \frac{clicks}{shows} = \frac{clicks + C_0}{shows + S_0},\ where\ C_0 = \bar{ctr} * S_0$$

Below on figure 2 we show some comparison of this prediction with the "baseline" prediction.

3.3 Sketch of inference in MAP framework

Another way to construct this estimation is applying Bayesian framework. Assume that we have some prior distribution on ctr , say $P_{prior}(ctr)$. Then if we have statistics $shows_b$ and $clicks_b$ for some banner b from A, then the posterior distribution for ctr_b is calculated as follows.

$$P_{posterior}(ctr_b) = P_{prior}(ctr_b) * {ctr_b}^{clicks_b} * (1 - ctr_b)^{shows_b - clicks_b}$$

Let $P_{prior}(ctr)$ belongs to the family of Beta distributions (i.e. $P_{prior}(ctr) = \frac{ctr^a * (1-ctr)^d}{B(a+1,d+1)}$). Then the respective posterior distribution will be Beta distribution with parameters $(a + clicks_b, d + shows_b - clicks_b)$.

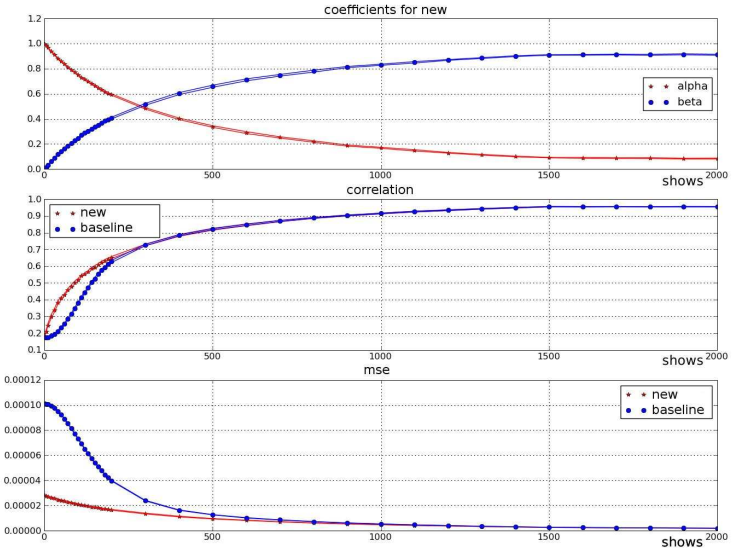


Figure 2: Top subplot exhibits form of optimal coefficients with respect to size of observations; middle subplot exhibits correlations of both predictions with realized clicks with respect to size of observations; bottom subplot exhibits mean square errors of both predictions with respect to size of observations.

And now it’s easy to show that the corresponding MAP estimation for ctr_b is

$$\hat{ctr_b} = \frac{clicks_b + a}{a + clicks_b + d + shows_b - ckickas_b} = \frac{clicks_b + a}{shows_b + a + d} = \frac{clicks_b + C_1}{shows_b + S_1}.$$

And again we have got the same form of the formula as in previous framework. But here all our prior information is encapsulated into two constants C_1 and S_1. There is question what values should be chosen for C_1 and S_1 (or, equivalentely, what Beta distribution should be used). But instead searching best values of unknown parameters we use some heuristic idea which appears from the inference in 3.2.

3.4 Further usage of the optimal form

All above justify using statistics of clicks and shows with some additional information as follows.

$$\hat{ctr_b} = \frac{clicks_b + D_0 * \tilde{ctr}}{shows_b + D_0},\ where\ \tilde{ctr}\ is\ an\ additional\ information.$$

Hereafter the form will be called as optimal form (OF).

What is more if we construct some ”static prediction” (i.e. without direct usage of statistics $clicks$ and $shows$, for instance by means of some regression like method on a number of text’s factors and user’s factors) as mentioned additional information $\tilde{ctr}$, which has some positive correlation R with corresponding random variate $click = \sum_{b\in A} click_b * show_b$ where $show_b = 1_{[b\ has\ been\ shown]}$, $click_b = 1_{[b\ has\ been\ clicked]}$, and has variance Σ such that $\Sigma < R^2 * var[click]$, then the similar inference like in expected mean square error minimization framework given $\alpha(shows) + \beta(shows) = 1$ gives us that $D_0 > S_0$. It is mean that if our additional information $\tilde{ctr}$ is better then the constant $\bar{ctr}$, , then we will trust former longer than last one (i.e. prediction $\hat{ctr_b} = \frac{clicks_b + D_0 * \tilde{ctr}}{shows_b + D_0}$ will converge to the ratio of $clicks_b$ to $shows_b$ slowly than $\hat{ctr_b} = \frac{clicks_b + S_0 * \bar{ctr}}{shows_b + S_0}$).

4 Reducing risk

To see different sides of new predictor we made a simulator. It takes a value of CTR and returns a plot, that shows probability of ad's bid growth in k times with respect to k. Looking at this plot, managers can decide how much risk we can allow.

The simulator shows that OF has higher probability of high prediction volatility for events with short banner history. It's not good for advertisers. They can lose their money and we can lose our clients.

We try to modify Optimal Form's values such that their variance on events with small banner history will coincide with variance on events with large banner history.

Let's select a number n_0 such that we will trust OF on events which has banner's history longer than n_0. For OF for each show number n we can calculate expectation M_n and standard deviation std_n of clicks' amount.

We want to satisfy the following conditions:

1. $(std_n)_{new} = std_{n_0}$ for every $n < n_0$
2. $(M_n)_{new} = M_n$

It is enough to take linear form of transformation. Thus, it assumes the following form:

$$OF_{new} = \frac{std_{n_0}}{std_n} * OF(clicks, shows) + \left(1 - \frac{std_{n_0}}{std_n}\right) * M_n, \; if \; n < n_0.$$

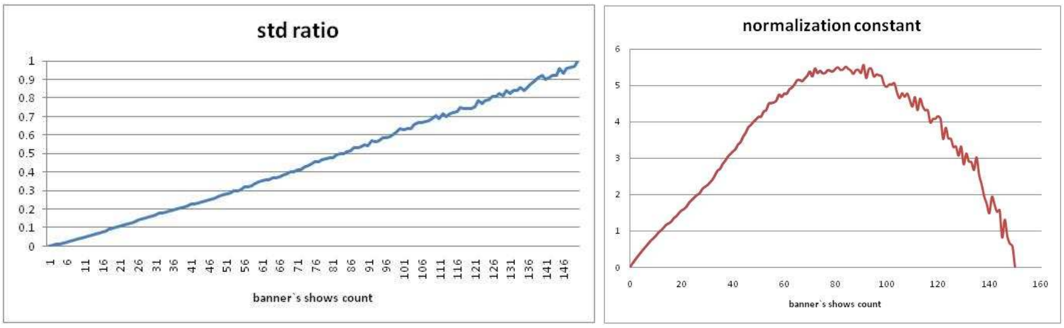


Figure 3: Left: Relationship $\frac{std_{n_0}}{std_n}$; Right: Normalization constant $\left(1 - \frac{std_{n_0}}{std_n}\right) * M_n$

After transformation OF_{new} has the same variance for all events which has banner history $n \leq n_0$. This step reduces the risks for advertisers associated with prediction volatility.

4.1 The experiment

We made an experiment where OF worked only for events with length of banner history in interval $[20, 80]$. There are about 14% of all events stay in this set. For other events we used baseline formula.

4.1.1 Results

In off-line experiment the new formula showed an increase in CTR at 1.9% and in log-likelihood at 1%. . On this basis, we decided to make an on-line experiment.

On-line experiment showed an increase in CTR at 1.53%.

Figure 4(left) shows that in the region where OF is used has an improvement with respect to old predictions.

It is evident (figure 4 right) that a new formula at this interval results a larger revenue.

Now let's consider the $CPC = \frac{money}{clicks}$.

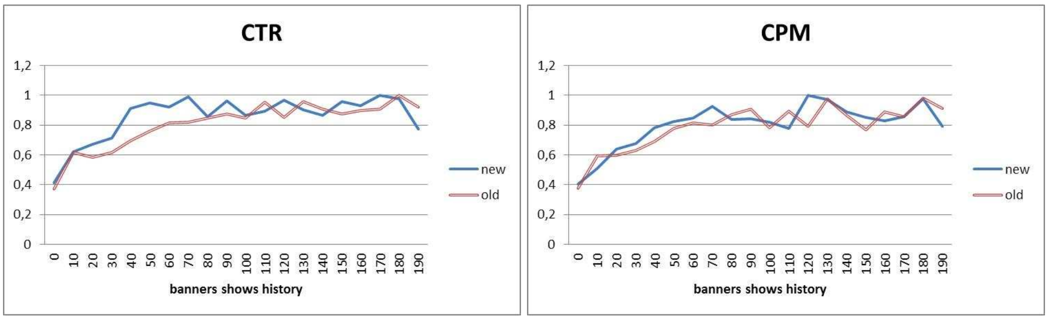


Figure 4: Left:CTR Right:CPM on the length of the history of banners

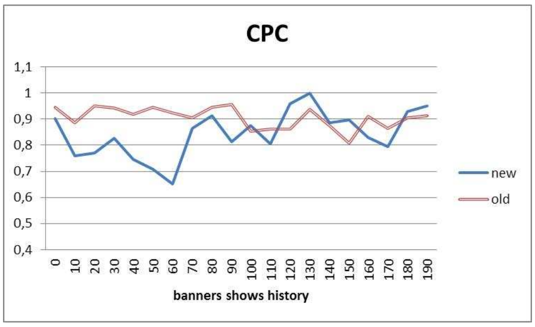


Figure 5: CPC on the length of the history of banners.

Figure 5 shows that in the OF's interval CPC became lower, hence more attractive. Thus we increase the click's traffic and the average click's cost decreased. This means that we give better terms for clients.

5 Conclusion

We described a new click through rate predicting formula. It was shown how to obtain this formula by two different approaches. Method of reducing risks at small banners history was introduced. Results of respective experiment were presented.

References

[1] T.Graepel, J.Q.Candela, T.Borchert, R.Herbrich, *Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine*

[2] K. Dembczynski, W. Kotlowski, D. Weiss *Predicting Ads' Click-Through Rate with Decision Rules*

[3] A. Ashkan, C.L.A. Clarke, E. Agichtein, Q. Guo *Estimating Ad Clickthrough Rate through Query Intent Analysis*

The Business Next Door: Click-Through Rate Modeling for Local Search

Suhrid Balakrishnan
AT&T Labs-Research
suhrid@research.att.com

Sumit Chopra
AT&T Labs-Research
schopra@research.att.com

I. Dan Melamed
AT&T Labs-Research
{lastname}@research.att.com

Abstract

Computational advertising has received a tremendous amount of attention from the business and academic community recently. Great advances have been made in modeling click-through rates in well studied settings, such as, sponsored search and context match. However, local search has received relatively little attention. Geographic nature of local search and associated local browsing leads to interesting research challenges and opportunities. We consider a novel application of relational regression to local search. The approach allows us to explicitly control and represent geographic and category-based neighborhood constraints on the samples that result in superior click-through rate estimates. Our model allows us to estimate an interpretable inherent 'quality' of a business listing, which reveals interesting latent information about listings that is useful for further analysis.

1 Introduction and Motivation

The rise of the Internet as an advertising medium has led to tremendous growth in computational advertising research and technology in the last decade. Since web users interact with the Internet very differently than they do with traditional media, such as, print, and television, many new technologies have been recently developed and deployed. Due to their large scale and nature (billions of ads served to hundreds of millions of users per day), computational/algorithmic techniques have been at the forefront of these technologies right from the beginning.

Extant computational advertising techniques typically address two modes of ad delivery. *Sponsored/Paid search* advertising places ads on a search engine relevant to a query [5, 2, 8]. *Context match* advertising places (contextually) relevant ads on a particular website [3, 1, 6]. Local search, as exemplified by sites such as Yahoo! Local, yellowpages.com, and Google Maps (local), is a separate problem domain that has received very little attention. Geography is the differentiating element of local search. Besides context, the query, and other factors normally considered in generic (non-local) computational advertising, successful local search advertising requires integrating location information. Also different is how the user interacts with local search. Indeed, there are essentially two main modes of online user interaction. In the first mode, which we term *look-up mode*, the user has a particular business in mind, and wishes to find more information about this business (an address, phone number, website etc.). As an example, a user may wish to find the hours of operation of their local post office. In the second mode, which we term *browse mode*, the user has a broad set of criteria that they would like satisfied by a particular business or, more generally, a set of businesses. In this mode of interaction, there is more of an exploratory/discovery flavor. For example, a user might be trying to decide on a restaurant in a particular neighborhood. We argue that any model for click-through rate in this domain should be cognizant of these modes of interaction, and that ignoring this or other related information is likely to be detrimental in terms of predictive accuracy.

In this work, we present a model that aims to capture these local search user interaction modes via incorporating "neighborhood" information while predicting the click-through rate. We choose to represent this information via neighborhood graphs, which are defined using *both* geographical location and business-category information together. The aim is that incorporating geographic locality helps with *look-up mode* queries, and incorporating topic/category information aids *browse mode* queries. We analyze the predictive performance of our model on a local search dataset, and show the importance of incorporating neighborhood information.

2 Relational Regression for Click-Through Rate Modeling

Our task is to model the click-through rates (CTR) of advertisements placed as a result of local search queries. The data we use for our experiments consists of aggregated daily clicks and impressions for a business listing/advertiser. In this dataset, the individual queries have been aggregated over as well, so we are not entirely in the sponsored search domain. Also note that the presence of geography means we are not quite in the context match domain either. In the notation we follow, the (log of the) CTR for a business listing i will be denoted by Y^i. As with other computational advertising settings, each business is associated with a vector of measurable factors or covariates X^i. For instance, these could be, *seniority of a listing*, i.e., when the business first established its account, *spend-category* of a listing, i.e., a category code indicating how much money a business listing paid to have its ads served etc. The predictive task is to accurately model the CTR for new/unseen businesses. This can be accomplished by modeling the dependence between the CTR of a business and its covariates. Our dataset consists of N such training sample pairs, which we collectively denote by $\mathbf{X} = \{X^i,\ i = 1 \dots N\}$, $\mathbf{Y} = \{Y^i,\ i = 1 \dots N\}$.

Whereas in the standard regression setting this task would likely be modeled assuming independence of the training samples, so that the estimate of Y^* for a test listing $*$ would treat two similar listings in the dataset independently, as argued in the previous section, we expect this approach is sub-optimal: when predicting CTR for a new pizzeria in a neighborhood, accounting for the CTR of a comparable pizzeria next door equally importantly as a pizzeria in the next town (which is what the independence assumption does) is clearly unsatisfactory. This ignores geographic neighborhood information. Also unsatisfactory is treating equally the effect of the next door pizzeria and hair salon which have similar measurable covariates, since this ignores business category information.

We address this particular nature of local search by applying relational regression [4]. At a high level, a directed neighborhood graph (on the businesses) is used to help specify probabilistic dependencies between listings in a formal manner, so that they are no longer treated as independent samples. In this work we examine various kinds of combinations of top-k neighborhoods that use both geographic locations and business categories (e.g., pizzeria, restaurant etc.). The neighborhood graph is then used to constrain the estimates of a (scalar) latent variable for each listing i, which we will denote by Z^i. In our relational dependency network model it is precisely these latent variables Z^i, that we make dependent on the latent variables of i's neighbors. Finally, the click-through rate Y^i for a business i is modeled as the sum (in log-space) of its *intrinsic* click-through rate I^i (a quantity that is dependent solely on the covariates of the business), *and* the neighborhood dependent latent variable Z^i.

$$Y^i = I^i + Z^i. \tag{1}$$

The variable I^i captures the covariate specific click-through rate signal and is modeled as a parametric function $f(\boldsymbol{\beta}, X^i)$. That is

$$I^i = f(\boldsymbol{\beta}, X^i) + \epsilon_1^i, \tag{2}$$

where β are the set of parameters, and ϵ_1^i is Gaussian error with zero mean and standard deviation σ_1^i. The variable Z^i captures the neighborhood specific click-through rate signal. Its value is a function of the Z^j's of the neighboring businesses. The dependency network showing these connections is in Figure 1. 'Unrolled', the network consists of a set of dense directed connections between the latent variables Zs, and associated sparse connections to each of the Ys. Intuitively, marginalizing over the latent variables, we would be modeling the Y and X samples jointly for all businesses. Instead, we force dependence between the businesses through the direct coupling of only the latent variables. Note that the plate notation slightly obscures the presence of cycles in this network (which is why this is not a Bayesian network). Indeed, there are many cycles, resulting from neighborhood connection links between the listings. For example, business A is a neighbor of business B ($A \in \mathcal{N}(B)$) and vice-versa ($B \in \mathcal{N}(A)$) implies a cycle of length two. We avoid such cycles by approximating the value of Z^i using a non-parametric function H of the neighboring Z^js

$$Z^i = H(Z_{\mathcal{N}(i)}) + \epsilon_2^i = \hat{Z}^i + \epsilon_2^i, \tag{3}$$

$\mathcal{N}(i)$ is the set of neighbors of i, $Z_{\mathcal{N}(i)}$ is the set of Zs of the neighbors, and ϵ_2^i is Gaussian noise with zero mean and standard deviation σ_2^i. Combining equations 1, 2 and 3, we model the click-through rate of a business i as

$$Y^i = f(\beta, X^i) + H(Z_{\mathcal{N}(i)}) + \epsilon^i, \tag{4}$$

where ϵ^i is Gaussian noise with zero mean and standard deviation σ^i.

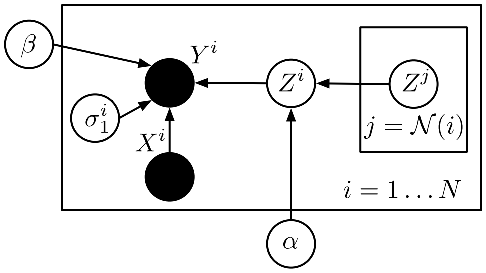


Figure 1: Dependency network for relation modeling of click-through rate for business listings. The figure uses plate notation for repetition. Observed values are shown as shaded nodes (the click-through rate Y and the listing covariates X). The cycles present in this graph can be most easily seen by considering the connections between the Zs in the unrolled version of the graph.

Neighborhood Graph Construction: A point of difference between our work and previous work on relational dependency networks, is that we do not try and explicitly learn the structure of the network between businesses (more accurately, their latent variables). Instead, we consider the structure of the network to be fixed and given. For ease of interpretation and computational efficiency, we consider k-sparse graph structures, where a business listing is connected only to its k closest neighbors (more on the particulars of the distance functions used in section 3). Further, we use non-parametric Nadaraya-Watson kernel regression for smoothly modeling the latent variable effects. In other words, for a business i we set $\hat{Z}_i = \frac{\sum_{j \in \mathcal{N}(i)} K_\alpha(d_{ij}) Z_j}{\sum_{j \in \mathcal{N}(i)} K_\alpha(d_{ij})}$, where K is a valid kernel function, d_{ij} is the distance between businesses i and j, and α is the kernel bandwidth parameter. In our experience, this model results in rich and sufficiently flexible dependency effects between the latent variables.

Inference: The standard approaches to approximate inference in relational dependency networks are mostly Bayesian, like ordered Gibbs sampling, or other graphical model based approximate inference procedures. In this paper, for scalability, practicality and due to our non-parametric neighborhood model, for which a probabilistic model isn't well defined, we pursue a different tack. Instead of computing a posterior distribution on the random variables, we will concentrate on obtaining (approximately) maximum likelihood point estimates of the parameters. Let us focus on the main model parameters for now, the regression coefficients $\boldsymbol{\beta}$ and the business specific latent variables $\mathbf{Z}$. We use the fact that conditioned on the latent variables $\mathbf{Z}$, the likelihood can be readily evaluated, and gradients of the likelihood with respect to the regression parameters can be obtained. Also, conditioned on the regression parameters, the latent variables can be optimized for. This intuitively leads to an EM-like scheme wherein starting from various random initializations for $\boldsymbol{\beta}, \mathbf{Z}$, these two phases are iterated until convergence to a (local) maximum. For scalability, we employ stochastic gradient descent for the optimization steps in updating $\boldsymbol{\beta}$ given $\mathbf{Z}$. We point out that using stochastic gradient descent makes it possible to use arbitrarily complicated regression models for $f(\boldsymbol{\beta}, \mathbf{X})$. Indeed, in this paper, the function f was a two layer neural network with 100 units in the first hidden layer, followed by a single output unit, giving the estimate of the covariate specific click-through rate signal. The final output unit does not have any non-linearity.

The above inference procedure can also be justified as inference in energy based models. Briefly, we can view the regression model and latent variable model as having two separate energy terms that need to be jointly minimized. Energy based learning is closely related to probabilistic graphical modeling and while space limitations prevent us from explaining further aspects of energy based modeling, we refer interested readers to [7] for greater detail.

Before moving to our experiments, we point out that by taking into account the geographic and category based neighborhood of listings, our model fits the local search paradigm well. Additionally, the latent variable Z provides a continuous effect that is estimated *conditioned* on the covariates X, which makes it interpretable and potentially useful. We can think of Z as modeling any residual click-through rate after taking into account the covariates. Thus, the Z estimates we obtain should allow us to compare businesses from two different click-through strata. As an example, by comparing their respective Z values, we can compare a plumber's listing to a pizzeria's. Thus in some sense this latent variable can be thought to represent an overall conditional 'quality' of a listing (in terms of click-through) and as such may be useful in the same way page-rank or other global listing rankings are used. This is an avenue of future work.

3 Data and Experiments

The data we use in our study consists of anonymized web traffic collected over a three month period from a popular local search website. The raw data we considered contained daily aggregated impressions (the event of showing a business listing), clicks and various features associated with the listing and how/where it was displayed. In order to study local effects, we chose to examine a high traffic geographic region around New York City.

We aggregated millions of records of raw data over the three months to create one single dataset to use for this analysis. Businesses with low impression counts over the period and those with missing covariate information were filtered out, resulting in our final dataset containing 2049 separate business listings. The covariates per business we considered were grouped in three different sets. Basic covariates, of which we had four, namely, the spending-based tier and priority level assigned to the business, the date when the business first joined the website service, and the average user rating of the business. The second set of covariates are related to the business category, and are simply the (multiple) categories that the business is listed under. This is represented in our data by a large sparse binary indicator matrix, with 2049 rows and 1658 columns. The columns correspond to 1658 listing categories. Finally, we also have geographic covariates, which simply consist of the latitude, and longitude of the business.

For our relational regression model we create neighborhood graphs based on combinations of geography and category features. Exploratory data analysis led us to using the Jaccard distance between the binary category feature vectors and great-circle distance for the geographic data. To create combination category + geography neighborhoods, a (λ, $1 - \lambda$) blend of the top rank score of sorted neighbors from each of category or geography were considered. The rank score we chose decays the rank by an exponentially down-weighted amount further in the position of the ranked list (reminiscent of NDCG), so that the "closer" neighbors matter much more than the "far-off" ones. These similarity graphs are then truncated at the top k position for the neighborhood graphs we use in our models.

To maximize the size of data for repeat experiments, we perform 10-fold cross validation to test our models. For each cross validation test fold, we also randomly sample one other fold as validation data for hyperparameter tuning and the remainder forms the training data (a train/validation/test data split ratio of 80/10/10). The evaluation measure we focus on is improvement in mean squared error. The values of the hyper-parameters α, η, were chosen by examining the validation set: α was set to 0.01, and η which denotes the learning rate of the neural network was initialized to 0.00001 and decreased by half after every 20 epochs. For the kernel regression on the Zs, we employ an exponential kernel, $K_\alpha(d_{ij}) = e^{-d_{ij}/\alpha}$.

4 Results

Our results for click-through rate estimation using relational regression are in Table 1, where we report the percentage improvements in log click-through rate mean squared error over a linear regression baseline. The click-through rate appears to be better modeled non-linearly, as can be seen by the improvements made using random forests and gradient boosted regression trees [9], on the same data. However, relational regression is the clear winner—our best relational models improve the baseline by around $\mathbf{30}\%$, that is, more than double the performance improvement by our best non-relational models (around $\mathbf{14}\%$). In addition, although most reasonable forms of neighborhood graph construction appeared to help the regression task, for this study, category information based neighborhoods appear to be more relevant to the signal. Adding geographic information to the neighborhood graph did not seem to aid click-through modeling. We conjecture that this is due to the density of businesses and ease of transportation in New York City.

Category Fraction, λ	**Neighborhood size, k**				
	5	10	20	50	100
0.00 (pure Geo.)	7.99	17.22	22.26	19.16	22.57
0.25	10.93	19.39	23.19	12.33	12.80
0.50	8.61	18.07	22.80	22.03	11.01
0.75	12.80	22.96	25.52	12.25	12.87
1.00 (pure Cat.)	22.73	29.32	30.49	**30.56**	29.86
Random Forests	6.16				
Gradient Boosted Trees	13.47				

Table 1: The mean percentage improvement in 10 fold cross validated log CTR mean squared error. The percentage is relative to a simple linear regression baseline model for the log click-through rate. The table shows relational regression results as a function of both the neighborhood size k, and the fraction of category to geography information used in the neighborhood, λ.

Rank	**Business Category**	
	Category neighborhood	Geographic neighborhood
1	Cocktail Lounges	Take Out Restaurants
2	Lodging	Italian Restaurants
3	Beauty Salons	Skin Care
4	Day Spas	Physicians & Surgeons
5	Hair Removal	Cocktail Lounges
6	Brunch & Lunch Restaurants	Lodging
7	Bars	Sports Bars
8	Sports Bars	American Restaurant
9	Restaurants	Bars
10	Printing Services	Caterers

Table 2: A comparison of the top categories for the businesses in our dataset. Middle column: categories ranked by aggregate Z for neighborhoods defined using only category information. Right column: shows the same except using pure geographic information for the neighborhood graphs.

Also interesting is an analysis of the estimated scalar latent variable Z. For the results that follow we fit the model to the full dataset, using the parameters and hyperparameters as determined by cross validation. Although noisy, a map of Manhattan showing the color coded value of the latent variable (red = high, blue = low) for all businesses (Figure 2, left column), allows one to recognize some popular areas, like mid-town and Greenwich Village. It also shows areas of low interest like the ones close to 1st and 2nd avenue just below mid-town, and the southern tip of Manhattan. Also shown in the right column of the same figure are two category specific Z maps. For the Arts category (top figure in the right column), we see a Broadway effect, with higher Z scores near the theater district and tapering off away from it. For the Food category (bottom right) we see relatively high Z scores in both the East and West Village areas, and lower scores in areas in between, and north of West Village.

We can also see interesting behavior by examining the top categories in terms of (normalized) average Z value (Table 2). We see that if we disregard geography in terms of neighborhood construction, we get a very different list than if we construct neighborhoods that are purely based on geographic information. In particular, the geography neighborhoods tend to be much more restaurant and evening-outing heavy. We conjecture that this may be due to the geo-neighborhoods being more suited for modeling *browse mode* queries which are more frequently used by tourists than local New Yorkers.

5 Future Work

Our results provide evidence for the applicability of relational regression models to the task of click-through rate modeling for local search, and we are currently examining other cities/regions to see if our hypotheses about locality and connectedness of regions hold. Other interesting avenues of future work are to relax the constraint on the network structure between business and learn it as well, and to extend the model to handle temporal effects.

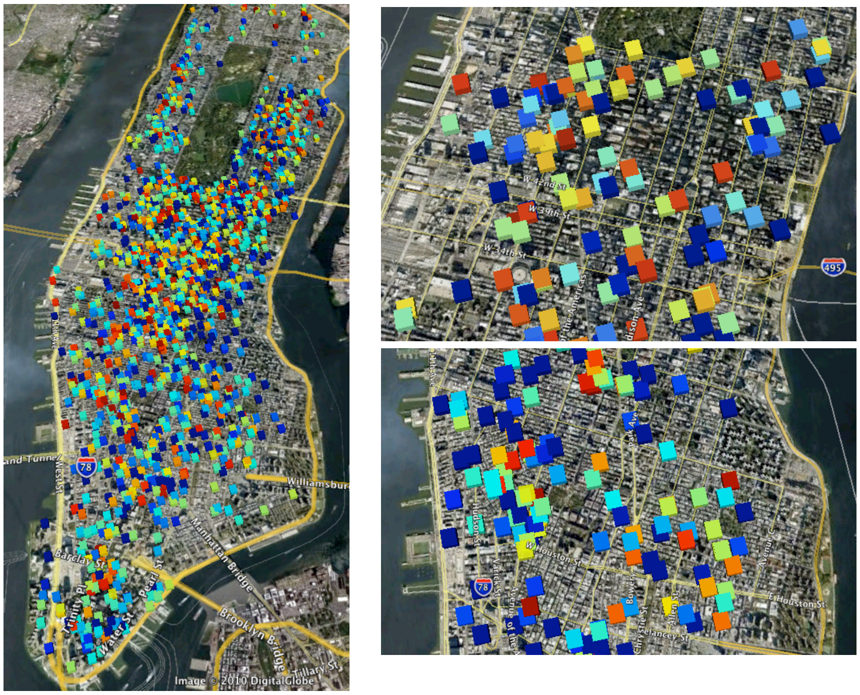


Figure 2: Map of the latent variable Z for the learned model using pure geographic information. The left panel shows all businesses in the dataset. The top right figure is for businesses in the category Arts and Entertainment, zoomed in to show mid-town. The lower right figure is businesses in the Food category, zoomed in around the East and West village areas. Colors range over the standard vibgyor color palette with indigo/blue on the low end of the scale and orange/red at the high end.

References

[1] 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 '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1003 – 1012, New York, NY, USA, 2008. ACM.

[2] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559 – 566, New York, NY, USA, 2007. ACM.

[3] D. Chakrabarti, D. Agarwal, and V. Josifovski. Contextual advertising by combining relevance with click feedback. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 417 – 426, New York, NY, USA, 2008. ACM.

[4] Sumit Chopra, Trivikraman Thampy, John Leahy, Andrew Caplin, and Yann LeCun. Discovering the hidden structure of house prices with a non-parametric latent manifold model. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 173–182, New York, NY, USA, 2007. ACM.

[5] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsofts Bing Search Engine. *ICML*, 2010.

[6] S. Gupta, M. Bilenko, and M. Richardson. Catching the drift: Learning broad matches from clickthrough data. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1165 – 1174, New York, NY, USA, 2009. ACM.

[7] Y. LeCun, S. Chopra, R. Hadsell, F. J. Huang, and M. A. Ranzato. *A Tutorial on Energy-Based Learning*. MIT Press, 2006.

[8] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 521 – 530, New York, NY, USA, 2007. ACM.

[9] Joseph Turian, Benjamin Wellington, and I. Dan Melamed. Scalable Discriminative Learning for Natural Language Parsing and Translation. In *NIPS '06: Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, BC.

Planning-based Approach for Optimizing the Display of Online Advertising Campaigns

Sertan Girgin, Jeremie Mary, Philippe Preux and Olivier Nicol
Team-Project SequeL
INRIA Lille Nord Europe, and LIFL (UMR CNRS), Université de Lille, France
name.surname@inria.fr

Abstract

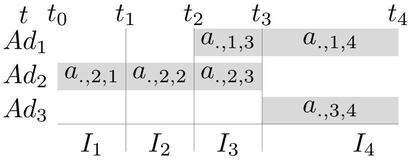
In a realistic context, the online advertisements have constraints such as a certain number of clicks to draw, as well as a lifetime. Furthermore, receiving a click is usually a very rare event. Thus, the problem of choosing which advertisement to display on a web page is inherently dynamic, and intimately combines combinatorial and statistical issues. We introduce a planning based algorithm for optimizing the display of advertisements and investigate its performance through simulations on a realistic model designed with an important commercial web actor.

1 Introduction and formalization of the problem

In this paper, we consider the problem of selecting advertisements in order to maximize the revenue earned from clicks in the "cost per click" economic model under different settings. Our goal is not to optimize any asymptotic behavior or exhibit algorithms that are able to achieve optimal asymptotic behavior, but rather to solve efficiently the practical problem that arises on a web site and involves certain degrees of uncertainty originating from various sources.

We define the problem as follows. At a given time t, there is a pool of K advertising campaigns denoted by K^t. Each campaign $Ad_k \in K^t$ is characterized by a tuple $(s_k, S_k, L_k, B_k, b_k^t, r_k)$ where k is the identifier of the campaign, s_k, S_k, L_k, B_k and r_k are its status, starting time, lifetime and total click budget and the revenue obtained per click respectively. $b_k^t \leq B_k$ denotes the remaining budget of the campaign at time t. The campaign lasts for L_k time steps and expects to receive B_k clicks during its lifetime. The status of an advertising campaign can be either: **scheduled** when the campaign will begin at some time in the future, **running** when the campaign is active (*i.e.* $S_k \leq t < S_k + L_k$), or **expired** when the campaign has ended (*i.e.* $S_k + L_k \leq t$ or $b_k^t = 0$). Only the advertisements of running campaigns can be displayed. The web server receives a continuous stream of visitors, each of which is assumed to be from one of N possible user profiles. The probability that the visitor belongs to a certain profile P_i is R_i with $\sum_{i=1}^{N} R_i = 1$. When a visitor visits the web site, a new "session" begins and we observe one or several iterations of the following sequence of events: (i) the visitor requests a certain page at time t (ii) the requested page is displayed to this visitor with an advertisement Ad_k embedded in it, (iii) the visitor clicks on the advertisement with probability $p_{i,k}$ where i denotes the user profile of the visitor; this probability is usually called the *click-through rate* (CTR), (iv) if there is a click, then the revenue associated with the advertisement r_k is incurred. After a certain number of page requests, the visitor leaves the web site and the session terminates. The objective is to maximize the total revenue by choosing the advertisements to be displayed "carefully". Since page requests are atomic actions, in the rest of the paper we will take a page request as the *unit of time* to simplify the discussion.

In the simplest case, we assume that (a) time horizon T is fixed, (b) the pool of advertising campaigns at each time step is given, (c) the visit probabilities of user profiles, R_i, and their click probabilities for each campaign, $p_{i,k}$, and the profile of each visitor are known. Note that, the visitor at time t and



(a)

$$\begin{array}{ll} \text{max} & \sum_{I_j \in \mathcal{A}^t} \sum_{Ad_k \in AI_j} r_k \mathbb{E}[\mathbf{Po}(p_{i,k} a_{i,k,j})] \\ \text{s.t.} & \sum_{Ad_k \in AI_j} a_{i,k,j} \leq \mathbf{Po}(R_i l_j), \forall\, 1 \leq i \leq N, I_j \in \mathcal{A}^t \\ & \sum_{i=1}^{N} \sum_{I_j \in IA_k} \mathbf{Po}(p_{i,k} a_{i,k,j}) \leq b_k^t, \forall\, Ad_k \in K^t \\ & \sum_{i=1}^{N} \sum_{Ad_k \in AI_j} a_{i,k,j} \leq l_j, \forall I_j \in \mathcal{A}^t \end{array}$$

(b)

Figure 1: (a) The timeline divided into intervals and parts. $Ad_{1,3}$ are in scheduled state at time t_1, and Ad_2 expire after t_3. I_j denotes the j^{th} interval $[t_{j-1}, t_j]$ and $a_{i,k,j}$ denotes the allocation for Ad_k for users belonging to profile R_i in interval I_j. The first index is dropped for the sake of clarity. (b) Stochastic formulation of the linear program. AI_j denotes the set of running campaigns in interval I_j, l_j is the length of interval I_j, and IA_k denotes the set of intervals that cover Ad_k.

whether he will click on the displayed advertisement or not are still unknown. Under this setting, given a visitor from profile P_i at time t, one possible and efficient way to choose an advertising campaign is to use the *highest expected value* (HEV) policy and pick the running campaign with the highest expected revenue per click, *i.e.* $\text{argmax}_{Ad_k \in K^t} r_k p_{i,k}$. Alternatively, one can employ the *stochastic expected value* (SEV) policy in which the selection probability of a running campaign is proportional to its expected revenue per click. As both policies exploit advertising campaigns with possibly high return and assign lower priority to those with lower return, they are expected to perform well if the lifetimes of the campaigns are "long enough" to ensure their total click budgets. However, even under some simple cases they may perform inferior to choosing an advertisement randomly at each step (see the example in [3] Sec. 2.1.1). In order to do better, it is compulsory to take into consideration the interactions between the advertising campaigns which materialize as *overlapping time intervals* over the timeline (Fig. 1 (a)); the problem then becomes finding an *allocation* of the number of advertising campaign displays in each interval such that (a) the allocation for a particular user profile is not over the capacity of the interval, (b) the remaining total click budgets are not exceeded, and (c) the total expected revenue is maximized. This corresponds to the maximization of a *linear objective function* subject to *linear inequality constraints*, which is a *linear programming* problem and can be solved efficiently; the detailed formulation and discussion can be found in [3]. The solution of the linear program at time t indicates the number of displays that should be allocated to each campaign for each user profile and in each interval, but it does not provide a specific way to choose the campaign to display to a user from a particular profile at time t. For this purpose, it is possible use the ratio of displays allocated to a particular campaign to the total allocation of advertising campaign displays for that user profile in the corresponding interval. One can either pick the campaign having the highest ratio, called the *highest LP policy* (HLP), or employ the *stochastic LP policy* (SLP) in which the selection probability of a campaign is proportional to its ratio. The linear program can either be solved at each time step or if this option is not feasible (*e.g.* due to computation time constraints) with regular periods or intermittently (*e.g.* when the budget of a campaign is met). In the latter case, the resulting allocation is used to determine the campaigns to be displayed until the next problem instance is solved by updating the allocated number of campaign displays as we move along the timeline and reducing the allocation of the chosen campaigns in the corresponding intervals. The complete algorithm is presented in [3] Fig. 4.

The static setting with full information has two sources of uncertainty: (a) the user profiles of visitors are drawn from a categorical distribution, and (b) each campaign display is a Bernoulli trial with a certain probability, which is known, and the result is either a success (*i.e.* click) or a failure. The aforementioned linear programming solution of the optimization problem focuses on what happens in the expectation. Following the resulting policy in different realizations of the random problem may lead to different total revenue that vary from its expected value (see the example in [3] Sec. 2.1.2). In reality, reducing this variability may also be important and could be considered as another objective. Note that, the expected number of visitors from user profile P_i during the timespan of interval I_j and the expected number of clicks that would be received if the campaign Ad_k is displayed $a_{i,k,j}$ times to the visitors from user profile P_i can be considered as random variables having Poisson distributions with parameters $R_i t$ and $p_{i,k} t$, respectively. Let $\mathbf{Po}(\lambda)$ denote a Poisson-distributed random variable with parameter λ. Replacing the corresponding terms in the linear program by the random variables, we obtain the stochastic optimization problem presented in Fig. 1 (b). The sum of independent Poisson-distributed random variables also follows a Poisson distribution with parameter equal to the sum of their parameters. Assuming that $\mathbf{Po}(p_{i,k}a_{i,k,j})$ are independent, the budget constraints can be written as $\mathbf{Po}(\sum_{i=1}^{N}\sum_{I_j \in IA_k} p_{i,k}a_{i,k,j}) \leq b_k^t, \forall\ Ad_k \in K^t$ which is equivalent to its linear program counterpart in expectation. The rationale behind this set of constraints is to bound the total expected number of clicks for each campaign, while at the same time trying to stay as close as possible to the bounds due to maximization in the objective function. Assume that in the optimal allocation the budget constraint of campaign Ad_k is met. This means that the expected total number of clicks for Ad_k will be a Poisson-distributed random variable with parameter b_k^t and in any particular instance of the problem the probability of realizing this expectation would be 0.5. In order to increase the likelihood of reaching the target expected total number of clicks, a possible option would be to use a higher budget limit. Let α_k be our risk factor and $\mathbf{Po}(\lambda_k)$ be the Poisson-distributed random variable having the smallest parameter λ_k such that $Pr(\mathbf{Po}(\lambda_k) > b_k^t - 1) \geq \alpha_k$; b_k^t and α_k are known, and λ_k can be found using numerical methods. If we replace b_k^t with λ_k in the budget constraint and solve the linear optimization problem again, the expected total number of clicks for Ad_k based on the new allocation would be greater than or equal to b_k^t and will have an upper bound of λ_k. Following the same strategy, one can derive new bounds for the user profile constraints and replace $R_i l_j$ terms with the corresponding parameters of the random variables.

So far, we have assumed that the visit probabilities of user profiles and their click probabilities for each campaign are known. In reality, these probabilities are hardly known in advance and have to be estimated. By noting that we can consider them as categorical and Bernoulli random variables, respectively, it is possible to estimate their value by using maximum likelihood or Bayesian maximum a posteriori estimation with conjugate priors of Beta and Dirichlet distributions (see [3] Sec. 2.1.3). As we will see in the next section, in the latter case choosing good priors may have a significant effect on the outcome. By estimating probabilities at each step (or periodically) and replacing the actual values with their estimates, we can determine allocations (optimal up to the accuracy of the estimations) and choose advertising campaigns to display. For maximum a posteriori estimates, the mode of the posterior distribution can be used as a point estimate and a single instance of the problem can be solved, or several instances of the problem can be generated by sampling probabilities from the posterior distributions, solved separately and then the resulting allocations can be merged (*e.g.* by taking their mean; in this case the final allocations will likely be not bound to the initial constraints). As in many online learning problems, one important issue that arises here is the need for balancing the exploitation of the current estimates and exploration, i.e. focusing on less-certain (*e.g.*, with higher variance) parameters; possible approaches are discussed in [3] Sec. 2.1.3.

In the more realistic dynamic setting, the time horizon is no longer fixed, and furthermore new campaigns may appear with time. We will consider two main cases in which either we have a *generative model* or not, which given a set of parameters and the current state can generate advertising campaigns during a specified time period. When a model is not available, only campaigns that have been revealed are known and they impose a certain maximum time horizon H_{max}. Although, it is possible to apply the proposed method and calculate the allocations for them, doing so would ignore the possibility of the arrival of new campaigns that may overlap and intervene with the existing ones; the resulting *long-term* policies may perform well if the degree of dynamism in the environment is not high. On the contrary, one can focus only on short or medium-term conditions omitting the known campaigns that start after a not-too-distant time H in the future. The resulting policies will be greedier as H is smaller and disregard the long-time interactions between the existing campaigns; however, they will also be less likely to be affected by the arrival of new campaigns (see the example in [3] Sec. 2.2). For such policies, choosing the optimal value of the planning horizon is not trivial due to the fact that it strongly depends on the underlying model. One possible way to remedy this situation would be to solve for a set of different planning horizons $H_1, \ldots, H_u = H_{max}$ (as the planning horizons differ, the structure of the optimization problems would also be different from each others) and then combine the resulting probability distributions of campaign displays, such as by majority voting. When a model is available, it can be utilized to compensate for the uncertainty in future events by allowing us to generate a set of *hypothetical* campaigns (for example, up to H_{max}), simulating what may happen in future, and include them in the planning phase. By omitting allocations made for these hypothetical campaigns from the allocation scheme found by solving the optimization problem, display probabilities that inherently take into consideration the effects of future events can be calculated. Note that, this would introduce bias to the resulting policies which can be reduced by running multiple simulations and combining their results as mentioned before.

2 Experiments

Our approach was tested on a toy-model designed with experts from Orange Labs, the research division of an important commercial web actor in France, to fit the real-world problem. We took care that each advertisement campaign has its own characteristics that more or less appeal to the different visits. The model assumes that each campaign A_k has a *base click probability* p_k that is sampled from a known distribution (*e.g.* uniform in an interval, or normally distributed with a certain mean and variance). As clicking on an advertisement is in general a rare event, the base click probabilities are typically low (around 10^{-4}). The click probability of a user belonging to profile P_i is then set to $p_{i,k} = p_k \gamma^{\mathbf{d}-1}$ where $\gamma > 1$ is a multiplicative coefficient and the random variable $\mathbf{d}$ is sampled from the discrete probability distribution with parameter n that has the following probability mass function $Pr[\mathbf{d} = x] = 2^{n-x}/(2^n - 1), 1 \leq x \leq n$. When n is small, all campaigns will have similar click probabilities that are close to the base click probability; as n increases, some campaigns will have significantly higher click probabilities for some but not all of the user profiles[1]. In the experiments we used two values for the γ parameter, 2 and 4; experts recommended use of the latter value, but as we will see shortly having a higher γ value may be advantageous for the greedy policy. The value of n is varied between 2 and 6. We opted to focus on core measures and therefore omit some of the extensions that have been discussed in the text.

We begin with the static setting with full information, and consider a fixed time horizon of one day assumed to be equivalent to 4×10^6 page visits. The distribution of user profiles is uniform and the budget and lifetime of campaigns are also sampled uniformly from fixed intervals. In order to determine the starting times of campaigns, we partitioned the time horizon into M equally spaced intervals (in our case 80) and set the starting time of each advertisement to the starting time of an interval chosen randomly such that the ending times do not exceed the fixed time horizon. The base click probability is set to 10^{-4}. We solved the optimization problem every 10000 steps. Fig. 2 (a) shows the relative performance of HLP policy with respect to the HEV policy for different values of the parameter n and budget for the case in which there is a single user profile and 40 campaigns with an average lifetime of $1/10^{th}$ of the time horizon; all campaigns have the same budget. We can make two observations, all other parameters being fixed HLP is more effective with increasing budgets, and the performance gain depends largely on the value of γ. For $\gamma = 4$, which is considered to be a realistic value by experts of the Orange Labs, and reasonable budgets the greedy policy performs well. A similar situation also arises when the number of campaigns is low, whereas increasing the number of user profiles favors planning as presented in Fig. 2 (b). Next, we consider longer static settings of over one week period with and without full information. The campaign lifetimes and their budget were more realistic (2-5 days, 500-4000 clicks). 7-9 new campaign are generated on a daily basis at the beginning of a run. We tested different values for the parameter n. There were 8 user profiles with equal visit probabilities. In this setting although HLP policy performs better than the greedy policy, the performance gain is limited (Fig. 2 (c)). While the greedy policy quickly exploits and consumes new advertisements as they arrive, HLP tends to keep a consistent and uniform click rate at the beginning and progressively becomes more greedy towards the end of the period (see [3] Fig. 10). Fig. 2 (d) shows the effect of the planning horizon; since we are not in the dynamic setting, using less information than available hinders the performance. Note that, this prominently depends on the degree of interaction between the campaigns and in this and other experiments we observed that being very far-sighted may not be necessary. Finally, we conducted experiments in the dynamic setting with partial information where the probabilities are estimated online. We employed ε-greedy exploration mechanism with different values of ε and maximum a posteriori estimation with Beta priors. The results in Fig. 2 (e) show that HLP can perform better than HEV, however for both policies the chosen hyper-parameters influence the outcome.

3 Related work

The oldest reference we were able to spot is Langheinrich et al. [6] who mix a linear program with a simple estimation of CTR to select advertisements to display. In this work, no attention is paid to the exploration/exploitation trade-off and more generally, the problem of the estimation of the CTR is

[1] Note that, the number of such assignments will be exponentially low; for fixed γ, the number of campaigns with click probability p will be twice that of with click probability γp. This allows us to model situations in which a small number of campaigns end up being popular in certain user profiles.

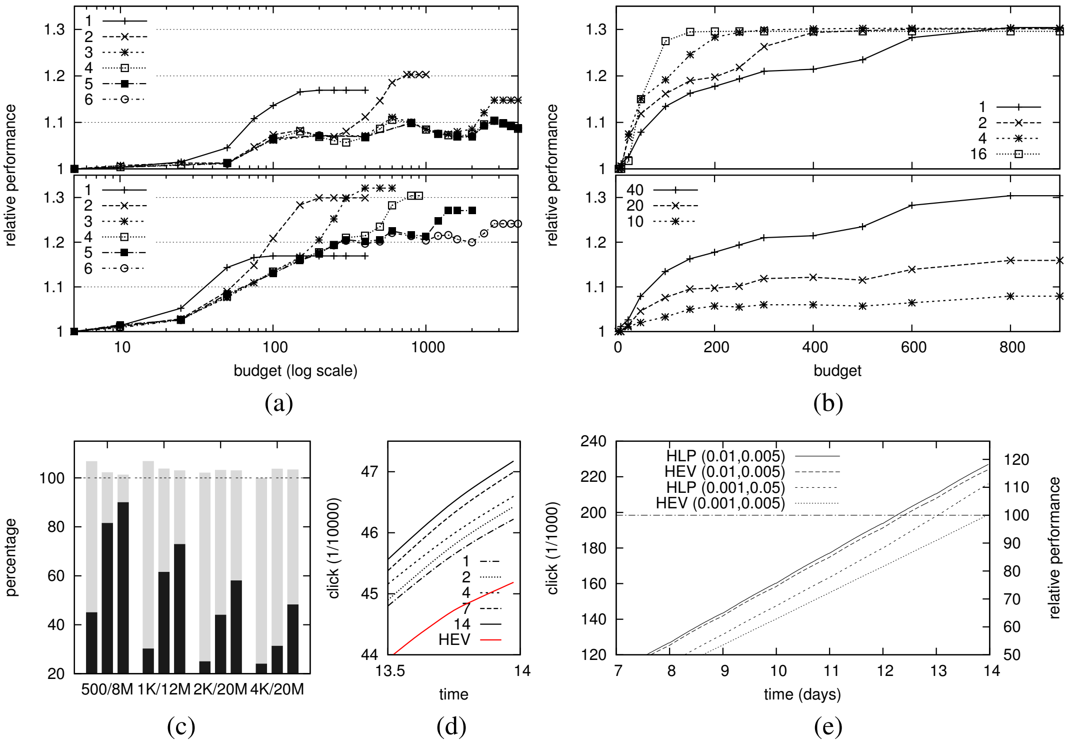


Figure 2: (a) The relative performance of the HLP policy with respect to the HEV policy for different values of n under the static setting with one profile and 40 campaigns. γ is 2 (bottom) and 4 (top). (b) The effect of the number of user profiles (top) and campaigns (bottom) for $n = 2$, $\gamma = 4$ and other parameters are kept constant. (c) The performance of the random (dark gray) and the HLP (light gray) policies with respect to the HES policy under the 7 days static setting for different budget (500 to 4000), lifetime (2-5 days) and n values. The three sets of bars in each group corresponds to the case where $n = 2, 4$ and 6 in that order. (d) The effect of horizon (1, 2, 4, 7, 14 days) in the 14 days static setting with full information. Bottom line shows the HEV policy. (e) The performance of HEV and HLP algorithms in the dynamic setting with partial information using ε-greedy exploration. The numbers in paranthesis denote the values of the parameter of the Beta prior and ε.

very crudely addressed. Abe and Nakamura [1] introduced a multi-arm bandit approach to balance exploration with exploitation under unlimited resources and with a static set of advertisements. This was later improved in [11] where they address the problem of multiple advertisements on a single page, and the exploration/exploitation trade-off using Gittins indices. Ideas drawn from their work on multi-impression may be introduced in ours to deal with that issue.

Aiming at directly optimizing the advertisement selection, side information is used to improve the accuracy of prediction in several recent papers [4, 5, 8, 12, 13]. However, all these works do not consider finite budget constraints, and finite lifetime constraints, as well as the continuous creation of new advertising campaigns; they also do not consider the CTR estimation problem. Very recently, Li et al. [8] focuses on the exploration/exploitation trade-off and proposes interesting ideas that may be combined to ours (varying ε in the ε-greedy strategy, and taking into account the history of the displays of an advertisement). Though not dealing with advertisement selection but news selection, which implies that there is no revenue maximization, and no click budget constraint, but merely maximization of the amount click, [2, 7] investigate a multi-arm bandit approach.

A rather different approach is that of Mehta et al. [10] who treated this problem as an on-line bipartite matching problem with daily budget constraints. However, it assumed that we have no knowledge of the sequence of appearance of the profile, whereas in practice we often have a good estimate of it. Mahdian and Nazerzadeh [9] tried then to take advantage of such estimates while still maintaining a reasonable competitive ratio, in case of inaccurate estimates. Extensions to click budget were discussed in the case of extra estimates about the click probabilities. Nevertheless, the daily maximization of the income is not equivalent to a global maximization.

4 Conclusion and future work

In this paper, we have provided insights on optimizing advertisement display, handling finite budgets and finite lifetimes in various settings within realistic computational time constraints. Our experimental results indicate that if there are few overlapping advertisements, or many advertisements with long lifetimes and good click rates, then we should be greedy. Between these two extreme solutions, one should consider the associated constraints. In particular, the lifetime of campaigns seem important. As future work, one possibility is to solve the problem from the perspective of the advertiser, *i.e.* help them to set the value of a click, and adjust it optimally with respect to the number of visitors (equivalent to a local sensitivity analysis of the LP problem). A more difficult issue is that of handling multiple advertisements on the same page where the correlation between the advertisements becomes important. Finally, we are also willing to draw some theoretical results on how far from the optimal strategy we are.

Acknowledgment This research was supported, and partially funded by Orange Labs, under externalized research contract number CRE number 46 146 063 - 8360, and by Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the "Contrat de Projets Etat Region (CPER) 2007-2013", and the contract "Vendeur Virtuel Ubiquitaire" of the "Pôle de compétitivité Industries du Commerce". Simulations were carried out using the Grid'5000 experimental testbed, an initiative from the French Ministry of Research, INRIA, CNRS and RENATER and other contributing partners.

References

[1] N. Abe and A. Nakamura, "Learning to Optimally Schedule Internet Banner Advertisements," in *Proc. of the 16th ICML*, 1999, pp. 12–21.

[2] D. Agarwal, B. Chen, and P. Elango, "Explore/exploit schemes for web content optimization," in *Proc. of the 2009 IEEE Int'l Conf. on Data Mining (ICDM)*, 2010, pp. 661–670.

[3] S. Girgin, J. Mary, P. Preux, and O. Nicol, "Advertising campaigns management: Should we be greedy?" Sep. 2010, INRIA Research Report RR-7388, http://hal.inria.fr/inria-00519694/en/.

[4] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "Efficient Bandit Algorithms for Online Multiclass Prediction," in *Proc. of the 25th ICML.* New York, NY, USA: ACM, 2008, pp. 440–447.

[5] J. Langford and T. Zhang, "The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information," in *20th NIPS*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. MIT Press, 2008, pp. 817–824.

[6] M. Langheinrich, A. Nakamura, N. Abe, T. Kamba, and Y. Koseki, "Unintrusive customization techniques for web advertising," *Computer Networks*, vol. 31, Jan. 1999.

[7] L. Li, W. Chu, J. Langford, and R. Schapire, "A contextual-bandit approach to personalized article recommendation," in *Proc. of the 19th WWW*, apr 2010.

[8] W. Li, X. Wang, R. Zhang, Y. Cui, J. Mao, and R. Jin, "Exploitation and exploration in a performance based contextual advertising system," in *Proc. of the 16th ACM KDD*, 2010.

[9] M. Mahdian and H. Nazerzadeh, "Allocating Online Advertisement Space with Unreliable Estimates," in *ACM Conference on Electronic Commerce*, 2007, pp. 288–294.

[10] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "Adwords and Generalized On-line Matching," in *Proc. of the 46th Annual IEEE FOCS.* IEEE Computer Society, 2005, pp. 264–273.

[11] A. Nakamura and N. Abe, "Improvements to the linear programming based scheduling of web advertisements," *Electronic Commerce Research*, vol. 5, no. 1, pp. 75–98, Jan. 2005.

[12] S. Pandey, D. Agarwal, D. Chakrabarti, and V. Josifovski, "Bandits for Taxonomies: A Model-based A pproach," in *Proc. of the 7th SIAM ICDM*, 2007.

[13] C.-C. Wang, S. Kulkarni, and H. Poor, "Bandit Problems With Side Observations," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 338–355, 2005.

Pricing Externalities in Real-Time Bidding Markets

Joseph Reisinger[*][†]
joeraii@cs.utexas.edu

[*]Department of Computer Science
University of Texas at Austin
Austin TX 78705

Michael Driscoll[†]
michael@metamarketsgroup.com

[†]Metamarkets Group
300 Brannan Suite 507
San Francisco CA 94107

Abstract

Online publishers rely on *real-time bidding* (RTB) markets to sell their remnant inventory and increasingly to command higher prices for "premium" content. However, content providers have lagged advertisers historically in the sophistication of their pricing models, as evidenced by the large increase in demand-side platforms without corresponding investment on the sell-side. Informed advertisers collect user-intent and demographic information in addition to the publisher context in order to score the relevance of impressions. The resulting differential impression pricing is only visible to publishers as a positive externality to revenue unless they collect the same audience targeting information. In this paper, we introduce a Bayesian hierarchical model of auction clearing price that can naturally account for the presence of *publisher-advertiser information asymmetry* and quantify its impact on price. Although the current model is simply exploratory, it suggests that richer models of dynamic floor pricing may improve publisher revenue.

1 Introduction

Real time bidding (RTB) markets have emerged as the preferred medium for ad networks and large advertisers to buy remnant inventory [3]. Individual publisher impressions are auctioned off in real time to participating advertisers, allowing them fine-grained control over audience targeting. In theory, publishers set floor prices in line with their view of the value of their inventory, and the degree of risk they must take on selling media to potentially unknown third-parties. Advertisers bid on the individual impressions, buying specific audience information, such as demographic, session history and intender status, from third-party data providers and demand-side platforms (DSPs), leading to information asymmetry between the demand- and supply-side. This presence of *informed bidders* amongst advertisers bidding on particular inventory causes *adverse selection* [6], with publishers raising floor prices across the board to avoid selling inventory at a perceived discount.

In this paper we explore the effects of informed bidders and information asymmetries between the supply- and demand-side in RTB markets and the resulting effects on empirical ad market microstructure. We posit that quantifying the effects of *information externalities* on informed bidding in aggregate can lead to more informed supply-side floor pricing, and hence increased publisher revenue, without the need for publishers to identify what impression level information is being specifically acted on by bidders. That is, the *presence* of differential pricing strategies, such as those employed by DSPs can be inferred directly from the bid price distribution.

Towards this end, we develop a mean-shift latent variable model in the context of linear regression to study publisher-advertiser information asymmetry, applying it to a large anonymized auction data set. The fundamental model assumption is that additional information available to a subset of informed advertisers, e.g. provided by DSPs, affects bid price additively, causing it to be overdispersed when compared to the baseline model. Hence, markets undergoing significant adverse selection due to information asymmetries will appear to publishers as additional clearing price dispersion. Although the underlying signals driving differential pricing may not be available on the supply side, publishers can still pool their auction data to estimate its economic impact.

In addition to the basic model, we discuss several extensions, including the potential to improve dynamic floor pricing mechanisms and produce more accurate estimates of marginal floor price. Ultimately, pooling sell-side data will help give publishers more fine-grained control over their inventory pricing and improve market efficiency.[1]

2 Mean-shift Mixture of Generalized Linear Models

We propose a simple generative model of auction clearing price p_i^a as a function of floor price p_i^f, publisher id x_i^{pid}, and a latent externality indicator z_i. Publishers set the floor price distributed over their inventory as a noisy signal of quality, forcing higher correlation between p^f and p^a. Advertiser willingness-to-pay is derived from a latent audience signal (unobserved information externality z_i) and site context.

Although the exact latent audience signal z_i cannot be reconstructed from the data, an aggregate estimate can be obtained by treating it as a latent variable in an overdispersed generalized linear model (GLM) framework,

$$\begin{array}{lcll}
\mathbf{w}|\Sigma_{\mathbf{w}} & \sim & \mathcal{N}(0, \Sigma_{\mathbf{w}}) & \text{(parameter weights)} \\
z_i|\mathbf{x}, \mu_0, \mu_1, \sigma_0, \sigma_1 & \sim & GMM(\cdot|\mathbf{x}, \{\mu_0 \leqslant \mu_1\}, \{\sigma_0 = \sigma_1\}) & \text{(latent group indicator)} \\
\boldsymbol{\alpha}|\sigma_\alpha & \sim & \mathcal{N}(0, \sigma_\alpha) & \text{(price mean-shift)} \\
p_i^a|\mathbf{x}_i, z_i, \mathbf{w}, \alpha & \sim & GLM(\cdot|\begin{bmatrix} \mathbf{x}_i \\ z_i \end{bmatrix}, \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\alpha} \end{bmatrix}) & \text{(regression)}
\end{array}$$

where $\mathbf{x}_i = \{p_i^f, x_i^{pid}\}$. This model combines a standard GLM with a two-component, equal variance Gaussian mixture model (GMM) indicating whether the auction price has been mean-shifted due to (unobserved) impression-level information. Such mean-shift mixture models are common in outlier detection, and can be used to model over-dispersion due to unobserved factors [2]. GLMs with latent factors can be fit using standard EM techniques; we adopt a Bayesian approach using Gibbs sampling [cf. 5].

Using this framework, we derive three particular models of p^a:

- **Externality-free** – Auction clearing price depends only on the observed variables,

$$p_i^a = w_0 + w_{0,x_i^{pid}} + (w_1 + w_{1,x_i^{pid}})p_i^f + \epsilon_i$$

  where $w_{0,\cdot}$ are intercept parameters and $w_{1,\cdot}$ are slope parameters. This model captures the contribution of the floor price and publisher id to the prediction of the auction clearing price.

- **Aggregate Externalities** – Audience pricing externalities are assumed constant across all publishers, hence some percentage of each publishers inventory experiences a latent mean-shift:

$$p_i^a = w_0 + w_{0,x_i^{pid}} + \alpha_0 z_i + (w_1 + w_{1,x_i^{pid}} + \alpha_1 z_i)p_i^f + \epsilon_i$$

  where α_0 and α_1 are additional intercept and slope parameters respectively for mean-shifted observations. This model captures additional price dispersion not accounted for in the baseline **externality-free** model, i.e. separating the systematic/floor-dependent portion of the bid from the additional unobserved audience segment signal.

- **Publisher-dependent Externalities** – Additional per-publisher coefficients are included to address *contextual* pricing externalities,

$$p_i^a = w_0 + w_{0,x_i^{pid}} + (\alpha_0 + \alpha_{0,x_i^{pid}})z_i + (w_1 + w_{1,x_i^{pid}} + (\alpha_1 + \alpha_{1,x_i^{pid}})z_i)p_i^f + \epsilon_i$$

  This model captures per-publisher deltas on the **aggregate externalities** model.

[1] For example, empirically, floor prices in RTB markets may be set too high, reflecting unreasonably high yield expectations for remnant inventory given the underlying market mechanics [8].

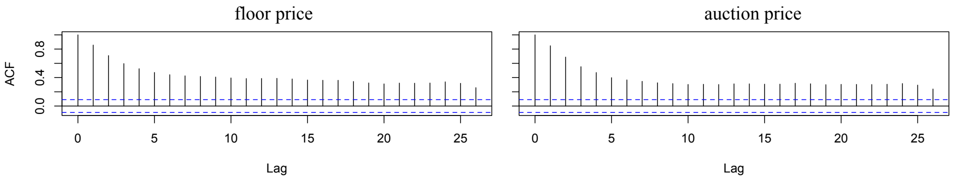


Figure 1: Hourly autocorrelation for $\mathbb{E}[p^f]$ and $\mathbb{E}[p^a]$ remains significant even for lags of a day or more, indicating time-of-day effects acting on auction pricing. We account for these effects before modeling other externalities by correcting for hourly residuals.

as well as variants that do not include publisher-specific effects (e.g. $w_{1,x_i^{pid}}$).

Examining the fits produced by each of these models allows us to determine the extent of differential pricing based on information externalities both across- and within publishers. If the model assigns $\mu_0 = \mu_1$ or $\alpha = 0$, then there is no additional price dispersion to be accounted for, and all price differentials must be due to publisher context. Furthermore, if $\alpha + w_0$ in the **aggregate externalities** model is less than w_0 in the **externality-free** model, then additional audience information has a *negative* impact on overall revenue, and vice-versa.

Finally, we note that this model is completely retrospective, and cannot be used to infer the existence of externalities acting on impression auctions before they take place; rather its utility lies in its use as a component for a differential floor-pricing model and demand-side weathervane.

3 Pricing Demand-Side Externalities

Dataset: 9M impression auctions from a real-time bidding market[2] with 5K publishers and 70 active advertisers[3] over a 2 week span during July 2010. Publisher ID and floor price are observed for all auctions; auction outcomes are observed in the form of the auction clearing price (second highest bid or floor price, whichever is higher). Since price data is typically closer to log-normal (no negative prices), we transform p^a and p^f into the log-domain. We also identify significant autocorrelations in both p^a and p^f on an hourly scale (Figure 1), indicating that time-of-day plays a role in price dispersion. To account for this potentially confounding effect, we fit a null model regressing on hour and adjust p^a and p^f according to the residuals.[4]

Table 1 provides statistics from the top publishers by volume from our auction sample, and highlights a wide variety in publisher strategies and inventory qualities. For example, all publishers except for 5 and 8 have dynamic floor pricing (nonzero floor price variance). Publisher 4 sets the highest floor price, maintaining inventory sell-thru of 10% and also has the highest divergence between floor and clearing price $\mathbb{E}(p_a - p_f)$. The publishers with the highest floor prices also have the highest correlation between p_a and p_f, indicating that advertisers are not willing to significantly overbid the floor. In general, publishers with low correlation between p_a and p_f are experiencing the most differential selection by informed bidders.

Table 2 summarizes the deviance[5] and most salient parameter coefficients for each of the models. The addition of the latent externality indicator z_i significantly improves the model fit, lending evidence for aggregate differential pricing based on unobserved information. However, the addition of per-publisher externality effects does not significantly reduce deviance beyond the aggregate model, indicating that advertisers may not be pursuing differential targeting based on publisher context. Rather, they may be primarily targeting cross-cutting demographics and user cookies. Figure 2 summarizes the contribution of each component of the **publisher-dependent externalities** model to the overall fit.

[2] Modified Vickrey (second-price) auction where the winning bidder pays the second highest price + $0.01.

[3] Winning at least one auction.

[4] The main results presented here do not depend on this correction.

[5] $D(y) = -2\left[\log(p(y|\hat{\theta_0})) - \log(p(y|\hat{\theta_s}))\right]$, where θ_0 are the parameters of the inferred model and θ_s are the parameters of a model with perfect fit (one parameter per data point)

Publisher	n	$N(p_a{>}p_f)$	$\mathbb{E}[p_f]$	$\mathbb{E}[p_a]$	$\rho(p_a, p_f)$	$\mathbb{E}(p_a - p_f)$
0	1.6M	192K	125± 75	151.3± 96.8	0.72	45.1 ± 79.9
1	1.5M	61.5K	35± 77	113.1± 110.4	0.53	66.1 ± 89.3
2	219K	37.9K	568± 504	462.9± 258.1	0.82	150.2 ± 199.0
3	174K	11.6K	111± 40	204.3± 167.2	0.51	100.1 ±161.1
4	138K	12.8K	734± 396	632.1± 254.8	0.87	151.0 ± 120.2
5	95K	35K	250±0	388± 130.0	-	138.7 ± 130.0
8	44K	26K	0± 0	129± 169.9	-	129.3 ± 169.9

Table 1: Examples of publisher impression price distributions for 7 of the top 10 publishers by volume. n is total impressions, $N(p_a{>}p_f)$ is the number of successful (cleared) auctions, $\mathbb{E}[p_f]$ is the average floor price (CPM in cents), and $\mathbb{E}[p_a]$ is the average auction clearing price. $\rho(p_a, p_f)$ is the Spearman's correlation between the floor and clearing price and $\mathbb{E}(p_a - p_f)$ is the expected lift over the floor price. Errors are standard deviations.

Model	$D(y)$	$\mathbb{E}[w_0]$	$\mathbb{E}[\alpha_0]$	$\mathbb{E}[w_1]$	$\mathbb{E}[\alpha_1]$
Aggregate Effects Only					
Externality-free	248736	3.21± 0.00	-	0.44±0.00	-
Aggregate Externalities	201443	2.57±0.00	1.32±0.01	0.39± 0.00	-0.09± 0.00
Publisher-Dependent Effects					
Externality-free	174768	0.96±0.02	-	0.87±0.00	-
Aggregate Externalities	110254	0.85±0.01	2.22±0.01	1.13±0.00	-0.40±0.00
Publisher-dependent Externalities	106122	2.79±0.02	1.92± 0.01	0.31± 0.00	-0.26± 0.00

Table 2: Inferred model parameters and model deviance. The *Aggregate Effects Only* models do not include per-publisher coefficients for the latent externality indicator z_i, while the models under *Publisher-Dependent Effects* do include such coefficients. $D(y)$ is the model deviance; w_0 is the log-price intercept; α_0 is the intercept delta when $z_i = 1$ (i.e. in the presence of a pricing externality); w_1 is the log-price slope; and α_1 is the is the externality slope delta. In the publisher-dependent effects case, reported values for the slope and intercept parameters are averaged across all publishers.

Across all models, the base price $w_0 + \alpha_0 z_i$ is significantly higher in the presence of externalities, as $\alpha_0 > 0$. Furthermore, $w_0 + \alpha_0$ in the externality model is greater than w_0 in the baseline model, indicating that additional information has a positive effect on auction revenue, at least for auctions that result in a sale.

The slope coefficient $w_1 + \alpha_1 z_i$ is lower in the externality models, as $\alpha_1 < 0$. This result makes intuitive sense: in auctions where externalities are found to affect bid price, clearing price is less sensitive to floor price (i.e. slope is near 0). In other words, the floor price, or publisher context, is less important as a signal of quality when advertisers have specific information about the particular impression (e.g. *auto-intender*, or *recently bought shoes*).

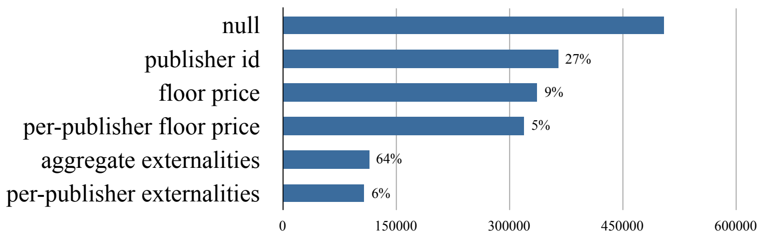


Figure 2: Residual deviance of linear model fit broken down over regression factors in the **publisher-dependent externalities** model. Publisher ID and latent mean-shift components induce the highest absolute reductions in deviance. Percentages show relative reduction in deviance with respect to the previous model.

4 Discussion

We have demonstrated the application of GLMs with a latent mean-shift parameter to quantifying the effects of information externalities and informed bidders on revenue in RTB markets. Such models are potentially useful to publishers interested in predicting the pricing dynamics of their remnant inventory.

4.1 Limitations

The main limitation of the proposed model is that it cannot predict *which* particular auctions are subject to information externalities, rather, it can only capture the aggregate effects on price dispersion retrospectively. However, publishers could potentially capture additional session and user features in order to make such predictions. Our model can then be used to gauge how much price dispersion is captured by such features.

Predictions from this model can be conflated with other causes of price dispersion, such as advertiser budgets fluctuating during the sample period, or seasonal effects on price.[6] In order to model demand-side externalities more accurately, it would be necessary to hold publisher, site context and the advertiser pool constant, observing price variation. However such controlled experiments are untenable in live markets.

4.2 Future work

Demand-side Modeling: Standard models of auctions assume bidders are endowed with their own private values over inventory and the auction clearing price is derived from the this set [cf. 4]. In this paper we have limited ourselves to modeling publisher effects, but could easily extend the analysis to include bidder preferences as well, bringing it more in line with traditional auction theory.

Supply-side Audience Targeting: There is significant market evidence for differential pricing based on audience targeting [cf. 8], and a natural consequent is for similar targeting to take place on the supply-side as well. Such dispersion due to dynamic floor pricing can be captured in our model.

Modeling Sell-through: Predicting sell-through (impressions sold) is also possible in the proposed framework, and is potentially interesting as unsold inventory may have undergone adverse selection due to information externalities.

Censored Models and Optimal Floor Pricing: In order to build models suitable for optimizing floor pricing it is necessary to have an estimate of what advertisers *would have* bid if a floor price were lower. Floor price can be treated as a dynamic *left-censoring*, where auction clearing price is not observed if it is below the floor price. Tobit regression can be used in place of linear regression in the presence of censored variables, and could potentially be used to reconstruct the full bid distribution [1]. Such models also allow straightforward temporal extensions [7].

Models of the full bid distribution would allow publishers to compute the *marginal floor price* and hence derive optimal floor pricing strategies. Theoretically, the optimal floor price is simply the second highest bid price (i.e. the market clearing price). However, in thin (demand-constrained) markets with few bidders and poor price-discovery, the floor price acts as a pseudo-bidder and can improve empirical supply-side revenue [9].

Pricing Risk: Finally, we envision extending our pricing models temporally in order to predict future spot market demand and volatility, key components in controlling publisher risk.

Acknowledgments

Thanks to Metamarkets[7] for making their RTB data available, David Soloff for helpful discussions on ad market microstructure and the anonymous reviewers for their detailed comments and for suggesting an interesting buy-side extension.

[6] We did not attempt to model temporal effects at time-scales longer than 24 hours as our sample period is too short to do do accurately.

[7] `http://www.metamarketsgroup.com`

References

[1] Siddhartha Chib. Bayes inference in the tobit censored regression model. *Journal of Econometrics*, 51(1-2):79–99, 1992.

[2] R. D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Chapman and Hall, 1982.

[3] LLC DeSilva + Phillips. Getting real ad exchanges, RTB, and the future of online advertising. `http://goo.gl/EGXe`.

[4] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[5] Lauren A. Hannah, David M. Blei, and Warren B. Powell. Dirichlet process mixtures of generalized linear models. In *Proc. of Artificial Intelligence and Statistics (AISTATS-10)*, 2010.

[6] Larry Harris. *Trading and Exchanges: Market Microstructure for Practitioners*. Oxford Press, 2003.

[7] Lung-fei Lee. Estimation of dynamic and arch tobit models. *Journal of Econometrics*, 92(2):355–390, October 1999.

[8] Mike Nolet. Price floors, second price auctions and market dynamics. `http://goo.gl/PgGR`.

[9] Michael Ostrovsky and Michael Schwarz. Reserve prices in internet advertising auctions: A field experiment. Research papers, Stanford University, Graduate School of Business, 2009.

Similarity Models for Ad Relevance Measures

Wen-tau Yih
Microsoft Research
One Microsoft Way
Redmond, WA, USA
scottyih@microsoft.com

Ning Jiang
Microsoft AdCenter
One Microsoft Way
Redmond, WA, USA
ninjian@microsoft.com

Abstract

Measuring the relevance between the query and paid search ads is an important problem to ensure the overall positive search experience. In this paper, we study experimentally the effectiveness of various document similarity models based solely on the content analysis of the query and ad landing page. Our approaches focus on two different aspects that aim to improving the document representation: one produces a better term-weighting function and the other projects the raw term-vectors to the concept space. Both models are discriminatively trained and significantly outperform the baseline approach. When used for filtering irrelevant ads, combining these two models gives the most gain, where the uncaught bad ads rate has reduced 28.5% when the false-positive rate is 0.1.

1 Introduction

Paid search advertising is the main revenue source that supports modern commercial search engines. When a user issues a query to a search engine, the search result page consists the organic links, as well as short textual ads on the mainline and sidebar. Although from the user's perspective, both organic results and paid search ads should respond to the search intent and provide relevant information, their generation processes are very different. While presenting relevant Web pages should be the only objective of the retrieval algorithm behind organic search, ad selection is heavily influenced by the market behaviors of advertisers. Generally, advertisers create short textual ads with a list of bid keywords and specified matching schemes. Only ads with keywords matching the query have the chance to enter the auction process, which decides the final ads to show.

Although advertisers should select keywords that are highly relevant to their ads and landing pages, naively trusting the auction and matching mechanism could allow showing irrelevant ads due to different reasons. For example, inexact matching schemes (e.g., *phrase* and *broad* match) only partially match the bid keyword to the query, which may be semantically distant. Another example is that adversarial advertisers may game the system by bidding on lots of cheap but irrelevant keywords to increase the traffic to their sites with relatively low cost. To ensure satisfactory user experience, it is thus important to have an ad relevance judgment component in the ad selection pipeline. Such component can be used simply as an ad filter, which only allows the ads with high relevance scores entering the auction process. More ambitiously, a good ad relevance measure can be used to replace the cumbersome keyword–query matching scheme and increase the ad coverage by selecting more ads to participate in the auction.

Previous work on ad relevance follows the vector space model paradigm in information retrieval and focuses on constructing suitable term-vectors representing the query and ad-text. For example, Broder et al. [2] leverage the search engine and use the pseudo relevance feedback technique to expand queries. Choi et al. [3] further enhance such approach by incorporating content from the ad landing page when creating term-vectors for the ads. The relevance function of a pair of query and ad is simply the cosine similarity score of the two corresponding vectors. The main advantage of this vector space approach is its computational efficiency when handling large-scale data in real-time. For instance, ad selection beyond keyword matching can be done via inverted index, pre-built offline on query and ad vectors. The content-based similarity score can also be combined later with other signals (e.g., advertiser reputation or user click features) to evaluate the relevance of ads from a pre-selected and much smaller candidate set.

In this paper, we aim to *learn* a better vector representation of queries and ads from simple content analysis so that a pre-selected similarity function (e.g., *cosine*) can become a reliable relevance measure. Instead of exploring various unsupervised heuristics of term weighting and selection in previous work, we exploit the annotated pairs of queries and ads and adapt two recently proposed vector learning approaches. The first approach, TWEAK [10], provides a simple means to incorporate various term-level features and results in a much better term-weighting function compared to using fixed formulas like TFIDF. The second approach, *S2Net* [11], maps the original term-vector representation to a much smaller but dense concept vector space, which allows matching of semantically related words, and improves the relevance measure even after dimensionality reduction. While both approaches significantly outperform the TFIDF cosine baseline in our experiments on real-world data, the best result comes from the combination of these two, which reduces the false-negative rate by 28.5% when the false-positive rate is 0.1 in an ad filtering scenario.

The rest of this paper is organized as follows. Sec. 2 gives the problem definition and describes our approaches in detail. Our experimental validation is provided in Sec. 3. Finally, Sec. 4 presents some related work and Sec. 5 concludes the paper.

2 Problem & Approaches

Informally, the problem we are trying to solve can be described as follows. Assume that we are given a set of query–ad pairs with human relevance judgement as training data. The goal here is to learn a function that maps the query and ad to vectors, so that their similarity score (*cosine* in this paper) can be used as a good relevance measure – e.g., the score of an "excellent" query–ad pair will be higher than the "bad" ones. Because both queries and ads contain only a few words and may not provide enough content, we expand them first to "documents" as the raw input. On the query side, we applied the same query expansion method described in [8, 2]. Each query in our data set was first issued to the Bing search engine. The top 100 search result snippets are concatenated to form a corresponding "pseudo-document". On the ad side, we used its landing page. Taking advantage of the human judgement labels, we experiment with two new discriminative approaches to learn the vector representation from documents. One can be viewed as an enhanced term-vector generation process and the other produces a low-rank representation.

2.1 Learning Enhanced Term Vectors

The most widely used document representation for similarity measures is arguably the bag-of-words term-vectors. Suppose $\mathcal{V} = \{t_1, t_2, \cdots, t_n\}$ is the pre-defined vocabulary that consists of a set of all possible terms (e.g., tokens, words) that may occur in each document. The vector that represents a document d is then $[s_1, s_2, \cdots, s_n]^T$, where s_i is the weight of term t_i. Such vectors are typically very sparse as s_i is set to 0 when t_i does not occur in d. Otherwise, s_i is often determined by some fixed weighting formula such as TFIDF (e.g., $s_i = tf(t_i, d) \cdot \log(N/df(t_i))$, where N is the number of documents in the corpus). Because the term weights dictate the quality of the similarity function operating on the term-vectors, here we adapt TWEAK [10] to learn a better weighting function by incorporating more term-level information using our labeled data.

Suppose that each term t_i from document d is associated with a short feature vector, $(\phi_1(t_i, d), \phi_2(t_i, d), \cdots, \phi_m(t_i, d))$, where ϕ_j is the function that captures some term-level information, such as its position in the document or whether it is capitalized. The new term-weighting function is a linear combination of these features, namely $s'_i = tw(t_i, d) \equiv \sum_{j=1}^{m} \lambda_j \phi_j(t_i, d)$, where λ's are the model parameters. Because the human relevance judgement labels are defined on pairs of queries and documents, we cannot use the non-existent "correct" term-weighting scores to train the model. Instead, the difference between the label and similarity score based on the current model will be back-propagated to tune the parameters in each training iteration. More detail on the loss function and training procedure will be described in Sec. 2.3.

2.2 Learning Concept Vectors

When applying an inner-product like similarity functions such as cosine or the Jaccard coefficient, one major weakness of the term-vector representation is that different terms, regardless of how semantically related they are, will not be matched. As an illustrative example, two semantically very close term vectors {buy:0.3, pre-owned:0.5, car: 0.4} and {purchase:0.4, used:0.3, automobile:0.2} will have 0 cosine score because they do not contain any identical terms. Obviously, having a better term-weighting function will not fix this issue as long as the choice of *active* terms remains the same.

A common solution to this problem is to map the original term-vectors to some "concept space", so that semantically close words will be captured by the same concept [6, 7, 1]. Instead of using a generative model, we aim to learn the projection matrix discriminatively by adapting a newly proposed Siamese neural network model, S2Net [11]. Formally, given a set of term-vectors representing queries or ads, we would like to learn a matrix $\mathbf{A}_{n \times k}$ so that a term-vector $\mathbf{s} = [s_1, s_2, \cdots, s_n]^T$ will be mapped to a k-dimensional vector $\mathbf{A}^T\mathbf{s}$, where $k \ll n$. In this low-rank representation, the association between each concept element and the original terms is described in the corresponding column of $\mathbf{A}$. Although the functional form of this model, the matrix $\mathbf{A}$, is identical to the projection matrix used in latent semantic analysis (LSA) [6], the model generation process is completely different. By tuning the model (i.e., all the $k \cdot n$ entries in $\mathbf{A}$) to have a better similarity score from the concept vectors, such dimensionality reduction technique can in fact increase the performance significantly.

2.3 Model Training

Suppose $\mathbf{s}_q$ and $\mathbf{s}_a$ are the concept or term vectors of the query–ad pair (q, a). Then the cosine similarity score is $sim_{\mathbf{\Lambda}}(q, a) \equiv \frac{\mathbf{s}_q \cdot \mathbf{s}_a}{||\mathbf{s}_q|| \cdot ||\mathbf{s}_a||}$, where $\mathbf{\Lambda}$ denotes the model parameters (i.e., λ's in TWEAK and $\mathbf{A}$ in S2Net). Assume that we map the human judgement label of (q, a) to $y \in [-1, 1]$, then one simple loss function we can use is the mean-squared error, defined as $L_{\mathrm{MSE}}(y, sim_{\mathbf{\Lambda}}(q, a)) = \frac{1}{2}(y - sim_{\mathbf{\Lambda}}(q, a))^2$.

However, in either the ad filtering or ranking scenarios, the ranking order of the ads or query–ad pairs is usually more important than the absolute score. We therefore use a pairwise loss function for training the models instead. Given two pairs (q_1, a_1) and (q_2, a_2) where the former is annotated as more relevant than the latter, let Δ be the difference of their similarity scores. Namely, $\Delta = sim_{\mathbf{\Lambda}}(q_1, a_1) - sim_{\mathbf{\Lambda}}(q_2, a_2)$. The loss function L we use is: $L(\Delta) = \log(1 + \exp(-\gamma\Delta))$, where γ is the scaling factor that magnifies Δ from $[-2, 2]$ to a larger range. This loss function can be shown to upper bound the pairwise accuracy. It can also be regularized further by adding a term like $\frac{\alpha}{2} \sum_j \lambda_j^2$ or $\frac{\alpha}{2}||\mathbf{A}||^2$. In the experiments in this paper, γ is set to 10 and α is 0.01 for TWEAK. S2Net is regularized using a simple early stop scheme tuned based on the validation set. Optimizing the model parameters can be done using gradient based methods, such as stochastic gradient decent or L-BFGS.

3 Experiments

In this section, we present our experiments of applying the above similarity learning models in the ad relevance problem, including the data collection process, tasks and evaluation metrics, as well as the detailed results.

3.1 Data & Tasks

The similarity models discussed above are evaluated on a proprietary dataset made available by Microsoft AdCenter. The dataset consists of 12,481 unique queries that were randomly sampled from the Bing search engine logs. For each query, a number of top ads ranked according to their monetization values or cost per impression (CPI) are selected. Ads with higher CPI have higher chance to be shown to search users, and thus are more sensitive to classification error. Ads with invalid or unreachable landing pages are removed from the dataset. This results in a total number of 681,100 query-ad pairs in the dataset, of which 446,781 unique pairs of queries and landing pages are found. Note that duplicates exist because multiple ads may point to the same landing page.

Feature	Remark
TF	term-frequency
DF	document-frequency
Loc	the position of the first occurrence of the term
Len	the length of the document
QF	query log frequency
IsCap	whether any occurrence of the term is capitalized
InQry	whether the term is part of the query
InUrl	whether the term is part of the URL
InAnchorText	whether the term is in an anchor text
InHtmlTitle	whether the term occurs in the title
InMetaDescription	whether the term appears in the meta-description section
InMetaKeywords	whether the term appears in the meta-keywords section
Emphasized	whether the term is emphasized in some special fonts

Table 1: Term-level features used in the TWEAK model. The QF feature is the number of times the term is seen as a query in search logs during an 18-month period. The logarithmic values of TF, DF, Len and QF are also used as features. InQry is only used for the query side. InUrl, InAnchorText, InHtmlTitle, InMetaDescription, InMetaKeywords and Emphasized are only used on ad landing pages.

Each query-ad pair is then manually labeled using a scheme that describes the relationship between concepts (or sets). In this scheme, each document is regarded as a concept, and the relationship between two concepts is one of the five relations, namely *same*, *subset*, *superset*, *overlap*, or *disjoint*. In our experiment, when the task is a binary classification problem, pairs labeled as same, subset, or superset (23% of the dataset) are considered relevant, pairs labeled as disjoint (60% of the dataset) are considered irrelevant, and others (17% of the dataset) are ignored. When pairwise comparisons are needed in either training or evaluation, the relevance order is *same* $>$ *subset* $=$ *superset* $>$ *disjoint*.

The dataset is split into training, validation and test sets by queries. Ads selected for a particular query always appear in only one of these three sets. This avoids the same query-landing page pair being used in both training and validation/test sets, which would happen if one randomly splits query-ad pairs in the dataset. In our experiments, 40% of the queries are reserved for training, 30% for validation and the remaining 30% for testing. The validation set is used to tune some hyper-parameters such as the number of training iterations and the weight of the regularization term.

As mentioned in Sec. 2, we use the search snippets to form "pseudo-documents" for queries and the landing pages for ads as the raw input documents. Our vocabulary set contains 29,854 words and is determined using a document frequency table derived from a large web corpus. Only words with counts larger than a pre-selected threshold are retained. When applying the TWEAK model to learn the term-weighting function, the features we used are summarized in Tab. 1.

We test our models in two different application scenarios. The first is to use the ad relevance measure as an *ad filter*. When the relevance score of an ad is below a pre-selected decision threshold, this ad is considered not relevant to the query and will be filtered before going to the auction process. As this setting is close to the typical anomaly detection problem, we present the Receiver Operating Characteristic (ROC) curves of tested models to show the trade-off between false-positive (i.e., mistakenly removed good ads) and true-positive (i.e., filtered bad ads), as well as the corresponding AUC scores as the evaluation metrics. The second one is the commonly ranking scenario as used in organic search, where the ads with keywords that match the query are purely selected and ranked by their relevance score. In this scenario, we use the standard NDCG scores as the evaluation metric.

3.2 Results

We compare four different configurations in our experiments. Served as our baseline, *TFIDF* is the basic term-vector representation with the TFIDF weighting ($tf \cdot \log(N/df)$). TWEAK has exactly the same terms in each TFIDF vector, but the weights are determined by the linear function of features in Tab. 1 with model parameters learned from the training data. Taking these two different term-

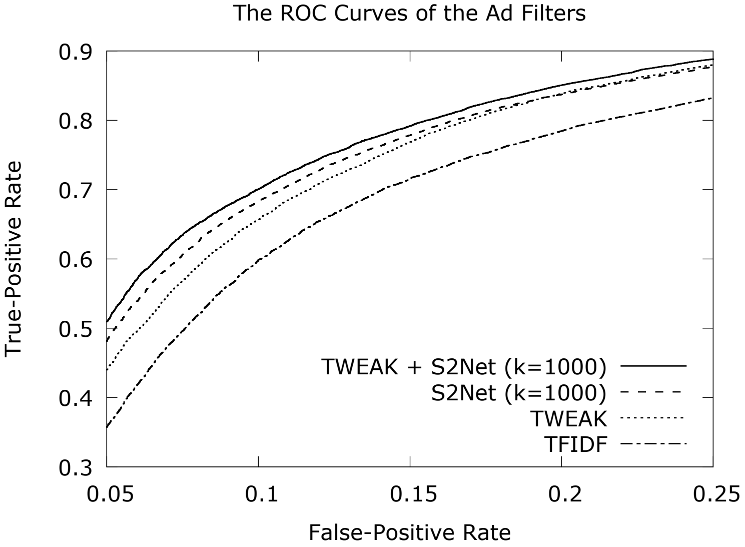


Figure 1: The ROC curves of four different vector representations when the corresponding cosine scores are used as ad filters. False-positive rate indicates the percentage of mistakenly filtered relevant ads and true-positive rate is the percentage of successfully filtered irrelevant ads.

	TFIDF	TWEAK	S2Net	TWEAK + S2Net
AUC	0.861	0.888	0.892	0.898
NDCG@1	0.825	0.859	0.855	0.870
NDCG@3	0.854	0.883	0.883	0.893
NDCG@5	0.876	0.899	0.901	0.909

Table 2: The AUC and NDCG scores of the cosine similarity scores on different vector representations. The dimensionality parameter k is 1000 for S2Net. Except for the NDCG scores between TWEAK and S2Net, the differences between any two methods in various metrics are statistically significant.

vectors as input, we applied S2Net to learn the projection matrices to map them to a k-dimensional space, denoted as S2Net and TWEAK+S2Net, respectively.

When the cosine scores of these vector representations are used as ad filters, their ROC curves (focusing on the low false-positive region) are shown in Fig. 1. It can be clearly observed that the similarity scores computed based on the learned vectors indeed have better quality, compared to the raw TFIDF representation. Among them, TWEAK and S2Net perform quite similarly, where S2Net has slight advantage when the false-positive rate is below 0.15. Not surprisingly, our best result comes from the combination of these two models. At the 0.1 false-positive rate point, TWEAK+S2Net can filter 28.5% more irrelevant ads compared with TFIDF.

Similar trend is also reflected on the AUC scores and NDCG numbers, presented in Tab. 2. S2Net has a higher AUC score compared to TWEAK, but is inferior in NDCG@1 and NDCG@3. TWEAK+S2Net is a clear winning approach, and has higher scores in both AUC and NDCG. Again, all the learning models result in stronger similarity scores than simply using TFIDF term vectors. All comparisons except for the NDCG scores between TWEAK and S2Net are statistically significant. For AUC, we randomly split the data into 50 subsets and ran a paired-t test between the corresponding AUC scores of two methods. For NDCG scores, we compared the DCG scores per query of the compared models using the paired-t test. The difference is considered statistically significant when the p-value is less than 0.01 after the Bonferroni correction.

Although for efficiency reason, ideally we would like the dimensionality of the projected concept vectors as small as possible. However, the quality of such representation usually degrades as well.

	TFIDF	PCA_{1000}	S2Net_{100}	S2Net_{300}	S2Net_{500}	S2Net_{750}	S2Net_{1000}
AUC	0.861	0.848	0.855	0.879	0.880	0.888	0.892
NDCG@1	0.825	0.815	0.843	0.852	0.856	0.860	0.855
NDCG@3	0.854	0.847	0.871	0.879	0.881	0.884	0.883
NDCG@5	0.876	0.870	0.890	0.897	0.899	0.902	0.901

Table 3: The AUC and NDCG scores of S2Net at different k (dimensionality) values. TFIDF and PCA ($k = 1000$) are used as baselines. The differences in AUC for any two methods, except for S2Net_{300} and S2Net_{500}, are statistically significant. For the NDCG scores, all S2Net models outperform TFIDF and PCA statistically significantly. The differences among S2Net_{300}, S2Net_{500}, S2Net_{750} and S2Net_{1000} are not statistically significant.

It is thus interesting to know the best trade-off point between these two variables. We conduct this study by using the raw TFIDF term-vectors as input for the S2Net model with various k values, and the results in terms of AUC and NDCG are shown in Tab. 3. In addition, we also compared the results with the commonly used dimensionality reduction technique, PCA. As can be found in the table, the performance of S2Net easily surpasses TFIDF when $k = 300$. As k increases, the performance improves quite consistently as well. Notice that even with $k = 1000$, PCA is not doing better than S2Net ($k = 100$), which uses only one tenth of the space, and is still inferior to TFIDF.

4 Related Work

Our approach on learning vector representation for similarity measures is very related to the work of distance metric learning [5]. As the computational complexity of learning a complete Mahalanobis matrix is at least $O(n^2)$, where n is the vocabulary size, directly applying them to the problems in the text domain is not practical. Although learning a low-rank matrix has been suggested [4, 9], our previous study has shown that the TWEAK/S2Net approach can perform better [11]. On the content analysis side, Choi et al. [3] used the cosine score to judge the ad relevance, but applied document summarization techniques to identify important portions in the ad landing page to construct the vector. Such information can easily be incorporated in our model and could potentially improve the performance.

5 Conclusions

In this paper, we explore the effectiveness of two recently proposed similarity models, TWEAK and S2Net, for measuring paid-search ad relevance. Both approaches aim to learn new vector representations of documents to improve the quality of the target similarity score (e.g., cosine) operating on the vectors. When used in the scenarios of ad filtering and ranking as relevance measures, the learned vector representations lead to significantly better results compared to the typical TFIDF term-vector construction. As the two approaches focus on different aspects and are complementary to each other, we found that combining these two methods produces the most performance gain.

The promising results from this initial experimental study trigger several interesting research direction for the future work. For example, the current combination approach treats the TWEAK and S2Net models separately and chains them in a sequential fashion. Training these two sets of model parameters could be a more natural approach to further enhance the overall model performance. On the feature side, improving the relevance measure by incorporating more information in the model, such as ad-text, advertiser reputation and deeper query and landing page analysis is also on our agenda.

References

[1] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] Andrei Z. Broder, Peter Ciccolo, Marcus Fontoura, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel. Search advertising using web relevance feedback. In *CIKM*, pages 1013–1022, 2008.

[3] Y. Choi, M. Fontoura, E. Gabrilovich, V. Josifovski, M. Mediano, and B. Pang. Using landing pages for sponsored search ad selection. In *Proceedings of the 19th World Wide Web Conference*, 2010.

[4] Jason V. Davis and Inderjit S. Dhillon. Structured metric learning for high dimensional problems. In *KDD*, pages 195–203, 2008.

[5] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 209–216, New York, NY, USA, 2007. ACM.

[6] Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[7] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57, 1999.

[8] Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th World Wide Web Conference*, 2006.

[9] L. Wu, R. Jin, S. Hoi, J. Zhu, and N. Yu. Learning bregman distance functions and its application for semi-supervised clustering. In *Advances in Neural Information Processing Systems 22*, pages 2089–2097. 2009.

[10] W. Yih. Learning term-weighting functions for similarity measures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, 2009.

[11] W. Yih and C. Meek. Learning vector representations for similarity measures. Technical Report MSR-TR-2010-139, Microsoft Research, October 2010.

Determining optimal advertisement frequency capping policy via Markov decision processes to maximize click through rates

James G. Shanahan
Independent Consultant
541 Duncan Street
San Francisco, CA 94131
James.Shanahan@gmail.com

Dirk Van den Poel
Ghent University
Belgium
Dirk.VandenPoel@UGent.be

Abstract

Digital online advertising is a form of promotion that uses the Internet and World Wide Web for the express purpose of delivering marketing messages to attract customers. Frequency capping is a term in digital advertising that means restricting (or capping) the amount of times (frequency) a specific visitor to a website or group of websites (in the case of ad networks) is shown a particular advertisement. Frequency capping is a feature within ad serving that allows the advertiser/ad-network to limit the maximum number of impressions/views a visitor can see a specific ad within a period of time. The advertiser or advertising network specifies a limit to the number of impressions you will allow per day, per week, or per month for an individual user. Frequency capping is often viewed as a key means in preventing banner burnout (the point where visitors are being overexposed and response drops) and in maintaining a competitive quality score (a core component of expected CPM-based ranking). Generally, the frequency capping policy for an ad is heuristically set by the advertiser or is determined heuristically by the ad network where the ad runs and is optimized for short term gain. In this paper we propose a data driven and principled approach that optimizes the life time value of site visitors. We propose to set frequency capping policies for different online marketing segments using Markov decision processes (MDP). Managing targeted marketing (customer relationship management) campaigns in this way can lead to substantial improvement in several business metrics such as click through rates and revenue. Though the current proposed approach lacks evaluation at the time of submission it is hoped to complete a study using this approach and present the results at the workshop.

1 Introduction

Digital online advertising is a form of promotion that uses the Internet and World Wide Web for the express purpose of delivering marketing messages to attract customers. Examples of online advertising include text ads that appear on search engine results pages, banner ads, in-text ads, or Rich Media ads that appear on regular web pages, portals or applications. Over the past 15 years online advertising, a $65 billion industry worldwide in 2008, has been pivotal to the success of the World Wide Web. That being said, the field of advertising has been equally revolutionized by new or transformed media sources such as the Internet, World Wide Web and more recently by the advent of social media sites and digital IP-TV.

This success has arisen largely from the transformation of the advertising industry from a low-tech, human intensive, "Mad Men" way of doing work (that were common place for much of the 20th century and the early days of online advertising) to highly optimized, quantitative, mathematical, computer-centric processes that enable highly targeted performance based advertising. In this spirit we propose a new data driven approach to tackling one aspect of digital advertising, that of creating frequency capping rules for ads shown to visitors (aka users, browsers, surfers) to a website. Currently, this is generally set heuristically based upon human background knowledge or experience. Frequency capping is a feature within ad serving that allows the advertiser/ad-network to limit the maximum number of impressions/views a visitor can see a specific ad within a period of time. The advertiser or advertising network specifies a limit to the number of impressions you will allow per day, per week, or per month for an individual user. For example, a rule of the form 3 views/visitor/24-hours means after viewing this ad 3 times, any visitor will not see it again for 24 hours. More elaborate capping policies are also possible and are generally composed of rules defined over multiple time periods. Frequency capping is often viewed as a key means in preventing banner burnout (the point where visitors are being overexposed and response drops) and in maintaining a competitive quality score (a core component of expected CPM-based ranking that has become a de facto standard for ad networks). Generally, the frequency capping policy for an ad is heuristically set by the advertiser or is determined heuristically by the ad network where the ad runs. And in some cases this carried out in an AB test manner where various policies are independently compared in a real world setting and assessed based upon business metrics such as CTR. Both approaches result in rules for all users and all ad types that focus on immediate short term gain, while ignoring the longer term value of having this visitor come back to this site, thereby leading to potentially great revenue opportunities. In this paper, we propose a data driven and principled approach to setting the frequency capping policy using Markov decision processes (MDP) that determines the best frequency capping policy for each user; it also focuses on the long term benefits of having such a visitor around. In summary this approach addresses two primary weaknesses of currently used practices: current approaches optimize for short term gains and largely ignore maximizing long term metrics (such as long term profits or CTRs), which is commonly known as the customer lifetime value (CLV); and in addition, current practices commonly set a policy at course levels for an advertiser thus leading to a one-size fits all type of capping policy that may be suboptimal.

The proposed Markov decision process that we adapt here builds on previous work carried out in the direct marketing field by Jonker et al. [1] where they treated the problem of maximizing long term profitability as joint optimization of both customer segmentation and marketing policy determination. They applied the approach to an offline marketing problem, that of requesting donations via direct mail solicitations, with great success. The work presented here differs on two fronts: first we explore different market segmentation strategies combined with feature selection algorithms as means of building homogenous groups of consumers; secondly we extend it to an online advertising problem setting, that of determining impression frequency capping rules for online marketing segments using Markov decision processes (MDP) such that we optimize a global marketing objective such as click through rate. To the best of the authors' knowledge this is the first application of MDPs to this particular problem.

This paper is structured as follows: In Section 2 we discuss our methodology. In Section 3 we describe the requirements for the experimental dataset. Section 4 discusses experiments while we close with conclusions in Section 5.

2 Methodology

In this study, similar in spirit to Jonker et al. [1], we adapt a population-based search paradigm called genetic algorithms to discover good site visitor segmentations and corresponding effective frequency capping rules for individuals assigned to those segments. Though this search algorithm is local in nature, sometimes leading to local optima, it generates surprisingly good results in practice in many domains [2]. This population-based search approach generates solutions to optimization problems, like the frequency capping problem presented here, using techniques inspired by Darwinian evolution (survival of the fittest), such as inheritance, mutation, selection, and crossover. Here fitness is based upon a combination of two measures: a bootstrap sampling estimate of a business metric, generally a proxy for revenue or profit, such as CTRs, for each segment; and a segmentation complexity component that penalizes segmentations with too many segments. This metric jointly evaluates the quality of the segmentation and the business value of optimally determined frequency capping rule. Individuals (candidate policies), composed of a customer segmentation and an optimal frequency capping rule for each segment, are constructed using two steps: step 1 segments customers into homogenous groups using clustering (either based on unsupervised and supervised methods); and step 2 determines the optimal frequency capping rule for each group using an MDP.

In direct marketing, the Recency, Frequency, and Monetary (RFM) variables are the most likely candidates as bases for site visitor segmentation (Bult and Wittink, 1996; Van den Poel, 2003; Viaene et al., 2001). The RFM variables measure consumer response behavior in three dimensions. The first dimension is recency, which indicates how long it has been since the customer has last responded. The second is frequency, which provides a measure of how often the customer has responded to received impressions. And finally, monetary value measures the amount of money (or clicks) that the customer has provided in response to the ad impressions. Various operationalizations of the RFM variables have been used in the literature. Here we plan to extend the traditional RFM variables with other variables that characterize the ad, the advertiser, and the user (or browser), and the publisher.

More formally, we define the RFM variables for our scenario as follows:

- Number of impressions of this ad without a click by user (Recency)
- How many times has this user been served ads from the topic of the ad before (Frequency)
- How long do I know this user
- How many times has this user been served before
- How many times did this user click on any ads (Monetary)
- CTR within the last 24, 48, 72, 168 hours over all displayed ads (Monetary)

We add the following variables to our segmentation basis also:

- Hour of day, Day of week
- Taxonomic distance from user category(ies) to ad category
- CTR of ad categories on the publisher category

One can imagine using many other variables in this basis. Our planned study will explore some of these candidates.

The goal of segmentation (for this paper) is to partition users into zones of self-similarity (homogenous groups) that can be described using a cluster center (the result of a k-means clustering [3] or SOM Clustering [4]) or a rule (a rule corresponds to one path from the root node to a leaf node of a learnt decision tree [5]). The segmentation step explores both unsupervised and supervised approaches to partition consumers into homogenous groups [6]. We consider unsupervised approaches such as k-means and supervised approaches such as decision-trees where the target variable is a business metric such as clicks or not clicks. Decision trees partition the set of users into subsets in which examples have similar values of the target variable, while clustering produces subsets in which examples have similar values of the descriptive variables. Variable selection and model parameter determination (such as the number of clusters), discretization of features (in the case of decision trees) will all be determined using genetic algorithms and will be expressed as part of the chromosome structure. Both components of the evaluation metric will naturally control for too many segments; this is discussed in more detail below.

The second step is concerned with determining an optimal frequency capping policy for each customer segment. The optimal policy is determined using an MDP as it models closely model the sequential nature of ad serving and optimizes for lifetime value of users. Modeling the problem this way avoids policies that optimize short term business gains and enables focusing on locating policies that optimize lifetime value of visitors. More formally, an MDP models the decision problem as inherently stochastic, sequential and fully observable. The solution to a MDP produces a policy or a "universal plan". A policy assigns to each state of the world (customer segment), an action (use a particular frequency capping rule to serve a particular ad) that is expected to be optimal over the period of consideration. An MDP can be defined as follows:

A Markov decision process is a tuple, $M = (S, A, T, R, H)$ where

- S is the set of all possible states (user segment in our case);
- A is the set of all possible actions (frequency capping rules);
- T is a transition function, $T : S \times A \rightarrow \Delta(S)$, which specifies the probability distribution over the next states given the current state and action;
- R is a reward function, $R : S \times A \rightarrow \Re$, which specifies the reward of performing each action from each state; and
- H is the period of consideration over which the plan must be optimal, also known as the horizon, $0 < H \leq \infty$.

The transition function as defined in a MDP is Markovian, that is, the probability of reaching the next state depends only on the current state and action, and not on the history of earlier states. Inclusion of the transition function allows MDPs to model and reason with non-deterministic (uncertain) actions. Furthermore, the horizon may be either finite or infinite. Here we assume the horizon is infinite, resulting in policy that is stationary over time.

Standard MDP solution techniques for arriving at an optimal policy revolve around the use of stochastic dynamic programming (Puterman, 1994) for calculation of the optimal policy. Bellman (Bellman, 1957), via his Principle of Optimality, showed that the stochastic dynamic programming equation, Equation (1), is guaranteed to find the optimal policy for the MDP. One standard MDP solution technique, called Value Iteration, involves iterating over Equation (1)-- calculating the expected value of each state -- until the value differential for each state reduces below a given threshold[1].

$$V^n(s) = \begin{cases} \max\limits_{a \in A} \left\{ R(s,a) + \gamma \sum\limits_{s' \in S} T(s' | s,a) V^{n-1}(s') \right\} & n > 0 \\ 0 & n = 0 \end{cases} \qquad (1)$$

where the function, $V^n : S \rightarrow \Re$ quantifies the long-term value, or reward, of reaching each state with n actions remaining to be performed.

Once we know the expected reward associated with each state, the optimal action for each state is the one which results in the maximum expected reward.

$$\pi^*(s) = \arg\max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s' | s,a) V^{n-1}(s') \right\} \qquad (2)$$

[1] Note that the value function provably converges over time.

In Equation (2), π^* is the optimal policy which as we mentioned before, is simply a mapping from states to actions, $\pi^* : S \rightarrow A$. We show the algorithm for computing the optimal policy in Figure 1.

```
Algorithm: ValueIteration( $M$ , $\varepsilon$ )
    Input:   $M$                 /* MDP to be solved */
             $\varepsilon$       /* max. allowed error in value of each state*/
    Output:  $\pi^*$             /* $\varepsilon$ -optimal policy */

    /* Compute the value function */
    Initialize $V'(s) \leftarrow 0 \quad \forall s$
    repeat
        for all $s \in S$ do
            $V(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s' | s,a) V'(s') \right\}$
        end for
        $\delta \leftarrow \| V - V' \|_\infty$
        $V' \leftarrow V$
    until $\delta < \varepsilon \frac{1-\gamma}{\gamma}$
    /* Obtain the optimal policy from the value function */
    for all $s \in S$ do
        $\pi^*(s) = \operatorname{argmax}_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s' | a,s) V(s') \right\}$
    end for
    return $\pi^*$
end algorithm
```

Figure 1: The Value Iteration algorithm for solving the MDP and computing the infinite horizon optimal policy.

Setting ad frequency capping can be very naturally captured by means of a Markov decision process. These MDP models describe the migration of an individual visitor in terms of transitions between states. A state records the segment that the visitor belongs to at a decision moment. Migration between the states will occur as a result of the ad serving rule and the response of the visitor in a time period between two decision moments. For each visitor the states are recorded at a number of decision moments. Let *a* be the impression decision at a decision moment for all visitors who are observed in state *s* (where there are *a=1, ..., A* actions and, *s=1, ..., S* states (or segments)). The transition probability to observe a migration from state *s* to state *t* after an impression decision a is denoted by $P_{s,t}(a)$. At each decision moment, action *a* triggers an immediate reward $r_s(a)$ for all visitors observed in state *s* at the decision moment. This reward is given by the response of the visitor till the next decision moment. The advertiser (a direct marketing firm) however may be more interested in long-term profits. Consequently, we model this reward as the long-term discounted reward as the objective for frequency capping optimization. Bitran and

Mondschein [7] show that this objective is closely related to lifetime value of the customers in mailing-based direct marketing studies. Other long or short term objectives can be used as desired. The Markov decision model can determine the optimal frequency capping policy by the well-known value iteration (described above) or policy iteration algorithm or by linear programming [8]. The parameters of the model that have to be estimated before the MDP optimization routine can start are the transition probabilities for each action *a* and the immediate reward in each state when action *a* is chosen.

The estimation of the parameters of the Markov decision model introduces an estimation error in the expected utility (e.g., CTR) for each capping policy. Most critical is the correct estimation of the transition probabilities $P_{s,t}(a)$. Sparse data sets are known to lead to unreliable estimates for transition probabilities. Similar to Jonker et al. [1], to address these problems in the parameter estimation, we use a bootstrap technique [9]. More specifically, we want to assess the stability of a solution (a partitioning scheme and a corresponding optimal capping policy) by calculating the bootstrap mean and standard deviation of the expected utility. This is accomplished by drawing bootstrap samples from the dataset, followed by assigning the visitors to segments using the partitioning scheme under investigation and optimizing the capping policy for this segmentation. In our setting, we only accept a solution as 'better' if it outperforms a given solution, i.e.,if (*mean–standard deviation*) of the new solution is higher than the (*mean–std.dev.*) of the old solution then the new solution is deemed better; this is calculated in terms of a business metric such as CTR.

The overall search to find an optimal policy (optimal frequency capping policy for an ad) for a site visitor is based upon genetic algorithms (GA) [10].There is a vast literature on GA, including studies on its theoretical and practical performance and many extensions of the basic algorithm [2]. For our application we consider the Simple Genetic Algorithm (SGA) as defined by Goldberg [2]. Although we realize that other, possibly more sophisticated GA formulations exist, we feel that SGA is best suited for our application because its simplicity will lead to a feasible running time of the algorithm. The genetic algorithm starts with a population of randomly selected solutions for the segmentation of the customers based on the segmentation variables. A solution can be denoted by the specification of partitions with regard to the segmentation variables. One then evaluates the goodness of each solution with respect to the current population. This evaluation consists of finding the optimal mailing policy for this segmentation and determining the expected performance of this policy. Solutions that result into a higher value are given more of a chance to 'reproduce' than others. For more details on GA see [10].

3 Data

In order to apply the proposed methodology we will need to collect data from ad serving preferably where different polices are explored across a visitor base; more precisely a broad set of policies (e.g., cap 24-hour frequencies at 1, 5, 10, 20 impressions) can be explored initially with more fine-grained policies (e.g., focus on capping 24-hour frequencies of 1, 2, 4, 5, 6, 7 if the 1 and 5 polices work best) being explored once more is learned from the initial grid search. Each user is assigned a user id that is stored locally on the visitors computer in a cookie (a text file) within the visitor's browser. Cookies are also used to store the impression count directly or indirectly. The logged data will consist of impressions and clicks records that detail the following:

- Date and time
- User-id of user that was shown ad j
- Ad id of the ad shown to user i
- Click or not
- Impression only
- Number of ads shown

- Category of ad j
- Category of user i
- Demographic features;
- Psychographic features such as intent of user (e.g., purchase);
- Web Browser's type
- Geographic features

Segmentation variables can be generated from the above log data. Though many features can be considered by our proposed GA-MD, it is very computationally intensive. To allieviate some of this computation, feature selection and discretization may need to be carried out in an independent first step using one-shot learning and clustering. The output of this step can then be used directly in building and evaluating each individual in the GA's population.

4 Experiments and Results

We plan to use the above GA-MDP based methodology to discover an optimal policy and compare it with incumbent approaches (which are largely heuristic-based) and report our results at the time of the workshop.

5 Conclusions

We have proposed a principled and data driving approach to setting frequency capping rules within the field of online advertising that focus on the lifetime value of site visitors. Generally, the frequency capping policy for an ad is heuristically set by the advertiser or is determined heuristically by the ad network where the ad runs. Our approach to setting frequency capping policies for different online marketing segments using Markov decision processes. Managing targeted marketing (customer relationship management) campaigns in this way can lead to substantial improvement in several business metrics such as click through rates and revenue. Though the current proposed approach lacks evaluation at the time of submission it is hoped to complete a study using this approach and present the results at the workshop.

This approach can also be extended to leverage more downstream user behavior such as dwell time on ads or associated landing pages or to transaction behavior (such as the amount of purchases a site visitor generates).

References

[1] J.J. Jonker, N. Piersma, and D. Van den Poel, "Joint Optimization of Customer Segmentation and Marketing Policy to Maximize Long-Term Profitability," *Expert Systems with Applications*, vol. 27, 2004, pp. 159-168.

[2] D. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, New York: Addison Wesley, .

[3] J.B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," Berkeley: University of California Press, 1967, pp. 281–297.

[4] T. Kohonen, *Self-Organizing Maps*, 2001: Springer, .

[5] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and regression trees*, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[6] B. De Reyck and Z. Degraeve, "Broadcast scheduling for mobile advertising," *Operations Research*, vol. 51, 2003, pp. 509-517.

[7] G.R. Bitran and S.V. Mondschein, "Mailing decisions in the catalog sales industry. Management Science," *Management Science*, vol. 42, 1996, pp. 1364–1381.

[8] S.M. Ross, *Introduction to stochastic dynamic programming, probability and mathematical statistics*, San Diego, CA: Academic Press, .

[9] B. Efron and R.J. Tibshirani, *Introduction to the bootstrap*, New York: Chapman and Hall, .

[10] J.H. Holland, *Adaption in natural and artificial systems*, Ann Arbor, MI: University of Michigan Press, 1975.

Predictive Client-side Profiles for Keyword Advertising

Mikhail Bilenko
Microsoft Research
Redmond, WA 98052, USA
mbilenko@microsoft.com

Matthew Richardson
Microsoft Research
Redmond, WA 98052, USA
mattri@microsoft.com

Abstract

Current approaches to personalizing online advertisements rely on estimating user preferences from server-side logs of accumulated user behavior. In this paper, we consider client-side ad personalization, where user-related information is allowed to be stored only within the user's control (e.g., in a browser cookie), enabling the user to view, edit or purge it at any time. In this setting, the ad personalization task is formulated as the problem of iteratively updating compact user profiles stored client-side to maximize expected utility gain. We compare the performance of client-side profiling to that of full-history server-side profiling in the context of keyword profiles used to trigger bid increments in search advertising. Experiments on real-world data demonstrate that predictive client-side profiles allow retaining a significant fraction of revenue gains due to personalization, while giving users full control of their data.

1 Introduction

Personalization has become a core component of modern web applications, where its uses vary from re-ranking search engine results to item recommendations in domains ranging from news to online shopping. Traditional uses of personalization center on customizing the output of information systems for each user based on attributes stored in their profile. Profile attributes may be explicitly or implicitly obtained, where explicit attributes are volunteered by the user or computed deterministically (e.g., user-submitted demographics, or IP-based location). Implicit user attributes are inferred based on logs of the user's prior behavior, e.g., of their past searching, browsing or shopping actions. A wide variety of personalization approaches have been proposed in recent years; notable examples include methods that leverage correlations between the behavior of multiple users (i.e., collaborative filtering), and approaches that use past behavior to assign users to one or more pre-defined categories (e.g., to behavioral targeting segments).

Raw behavior logs used to infer implicit user attributes are typically stored in the online service's datacenter (*server-side*), where they are processed to compute each user profile in a compact representation chosen for the application at hand. Examples of such representations include categories for behavioral targeting [3][21] and low-dimensional latent topics for collaborative filtering methods based on matrix decomposition [13]. The resulting profiles are used in subsequent interactions with the user to adjust the output of the application to user preferences.

Server-side aggregation is being increasingly questioned by consumer advocates due to the fact that it does not provide users the ability to view or control the data associated with them. As a result, there has been a rising interest in privacy-enhanced approaches to personalization, with one such approach being category-based profiles constructed and maintained entirely on the user's machine (*client-side*) for personalizing search results [18][20]. However, the trade-offs involved in moving user profiles client-side remain unclear.

In this paper, we formalize the problem of constructing client-side profiles based on the general framework of maximizing expected personalization utility. For many personalization utility functions that can be formulated as coverage problems, profile construction is a submodular optimization task, allowing efficient approximation algorithms with strong guarantees. We focus our attention on the utility of keyword profiles in search advertising accumulated via *bid increments*: keyword bid increases that allow advertisers to differentiate their campaigns for users with a strong interest in the topic. For this setting, we compare the performance of online, client-side profiling to full-history server-side profiling. Experiments on real-world data demonstrate that client-side profiling retains a significant fraction of server-side profiling revenue gains, while allowing users to opt out of server-side logging and gain full control of their behavioral history.

It is important to note that the presented approach is not a privacy-preserving ad delivery mechanism [7][6][19]: such mechanisms require installation of additional client software from users and significant changes to existing infrastructure and mechanisms from ad platforms. Our approach also does not aim to provide users statistical privacy guarantees such as those pursued by research in k-anonymity and differential privacy [14]. Instead, the goal of the paper is to describe a methodology for continuing to serve personalized advertising to users who have opted out of server-side logging, and to analyze the gap in personalization utility between client-side and server-side approaches in the context of search ads.

2 Advertising Personalization

Let $\mathcal{V}$ be the finite set of user behavior items, Φ be the domain of item descriptors, and $\mathcal{O} = \mathcal{V} \times \Phi$ be the domain of observed events. For example, in search advertising, $\mathcal{V}$ is the set of all advertiser-bid keywords that are matched to user queries, and $\Phi = \mathbb{R}^d$ contains vectors of features associated with a keyword being matched to a query. Then, every user query q can be represented as an observed event $o = (v, \phi)$ where $v \in \mathcal{V}$ is the most relevant ad keyword for the query, and $\phi \in \Phi$ is a vector of features capturing such event properties as the timestamp, keyword similarity to the query, user location, etc.

Let $\mathcal{H}$ be the domain of all sequences of observed events, and $\mathcal{P}$ be the domain of profile representations. A *profile construction* function $f: \mathcal{H} \to \mathcal{P}$ takes a sequence of events observed over a time period T, $H_T = (o_{t_1} \ldotp\ldotp, o_{t_n}), \forall t_i \in T$, and produces a user profile $p \in \mathcal{P}$. This definition can be trivially extended to include explicit or time-independent attributes such as demographic data.

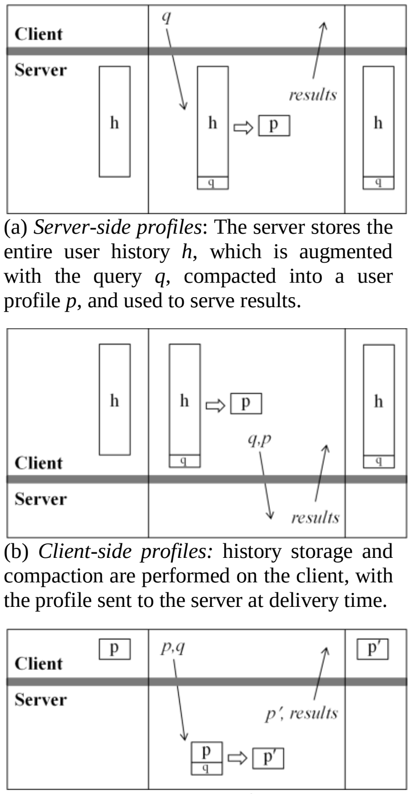
The objective of profile construction is to maximize some utility function that captures the increase in performance for the task at hand (the benefit of personalization). The utility function u is computed post-hoc by evaluating performance of the system over a future interval on a profile constructed during a preceding time interval, $u: \mathcal{H} \times \mathcal{P} \to \mathbb{R}$. The optimal profile construction function f^* maximizes expected utility: $f^* = \arg\max \mathbb{E}[u(H_{T^+}, f(H_{T^-}))]$, where the expectation is computed over the probability distribution of behavior across all users, while T^- and T^+ are time intervals over which the profile is constructed and used, respectively.

A number of utility functions have been considered in prior work on personalization, e.g., measuring improvements in web search result accuracy has been performed via gains in average precision [20] or click probability prediction [3][5][21]. The value of information approach [9] provides a general probabilistic framework for computing the utility of personalization.

2.1 Keyword-based Profiles

Although the above formulation applies to arbitrary domains, we will now focus on the search advertising setting where both observed events $\mathcal{V}$ and profile representation $\mathcal{P}$ correspond to bidded keywords. Unlike display advertising, modern search and contextual ad platforms associate advertisements with bids on individual keywords, which are then matched against queries or page content (either exactly or approximately). Hence, user profiles comprised of keywords can be naturally integrated with existing advertising systems and campaigns. To be useful in advertisement selection, ranking and pricing, such profiles contain the keywords in which a user has shown historical, as well as predicted future interest. By allowing advertisers to target users based on their keyword profiles, pay-per-click campaigns can be refined for users for whom they are likely to be more effective. For example, a diving equipment store may be willing to pay more for users who are predicted to have a long-term interest in specialized diving keywords, since they are more likely to purchase high-end items.

Requiring the ad keyword to be a part of a user's profile is limiting, since profile size is small, while the number of keywords that express any particular intent is large. E.g., an advertiser bidding on "*grass seed*" may wish to target users with corresponding long-term interests, but would then miss users whose profile contains "*lawn repair*". Note that the specificity of the interest rules out using lower-granularity approaches such as segments or categories. This problem is a key task for the ad selection phase of advertising delivery, where it is solved by constructing a weighted directed graph, G, with vertices representing all keywords and edges connecting keywords that should be matched. Building such graphs is a well-studied problem for online advertising [7][10][17], as it enables non-exact matching of keywords to queries (known as "broad" or "advanced" matching). Directed links allow e.g. "*Boston mortgage*" to point to "*mortgage*" to indicate that a user interested in the former will be interested in the latter, while the opposite is not generally true.



(a) *Server-side profiles*: The server stores the entire user history *h*, which is augmented with the query *q*, compacted into a user profile *p*, and used to serve results.

(b) *Client-side profiles:* history storage and compaction are performed on the client, with the profile sent to the server at delivery time.

(c) *Online client-side profiles*: The client stores only the compact profile, which is sent to the server along with the query. The server returns the updated profile and the results.

Figure 1: Comparison of server-side, client-side, and online client-side profiles.

Given a user profile of k keywords, $P = \{w_1, .., w_k\}$, we consider a current context (query or webpage) keyword w to *match* the user profile if it is either contained in it, or is a direct neighbor of at least one of the profile keywords. Because utility is typically an additive function of individual context matches, finding the optimal profile is an instance of the maximum coverage problem: selecting a fixed-size set of profile keywords maximizing a set-based objective function. While the problem is NP-hard in general, personalization utility being submodular guarantees that the greedy algorithm of Nemhauser et al.[14] produces a $(1 - {}^{1}/_{e})$-approximate solution.

3 Online Client-side Profiles

The general definition of profile construction in Section 2 presumes that the profile is constructed based on the complete history of user behavior over the previous time period, H_{T^-}. Currently, this history is stored server-side and updated as necessary. To give users full control over their information, the history and profile must be stored on the client, potentially incurring prohibitive local storage costs. Profile construction would happen either on the client (requiring additional browser components to be installed to enable computation in the browser [7][6][19], presenting a significant barrier to wide adoption) or the server (requiring the history to be communicated to the server whenever the profile is to be updated, thus incurring significant communication overhead).

Given these concerns, we consider the scenario where the history is not stored, and the (relatively small) user profile is stored on the client. The compact profile is sent to the server along with the current context (query or webpage id). The profile is then utilized on the server, updated and returned along with the ads served. This scenario is supported by current web browsers natively via cookies.

Updating client-side profiles online changes the problem from one of constructing a profile given the full user history to one of revising the present profile based on the current context. The corresponding profile construction function for updating the profile on per-event basis is then defined as $f_{online}: \mathcal{O} \times \mathcal{P} \rightarrow \mathcal{P}$, with the profile update at i^{th} observation computed as $p_i = f_{online}(p_{i-1}, o_i)$.

The corresponding utility, u_{online}, aggregated over interval T^+ is then computed as $u_{online}(p, H_{T^+}) = \sum_{o_i \in H_{T^+}} u(p_{i-1}, o_i)$ where p_o is the initial empty profile.

To alleviate the myopia suffered by the incremental update, we augment the profile with a cache of m recently seen keywords which are not a part of the profile, yet are retained alongside it client-side. Selecting the optimal k profile keywords from the up to $k+m+1$ known keywords is still submodular and may still be approximated using the greedy algorithm.

4 Method

We apply online client profiles to the *bid increment* setting, where advertisers are given the option to have their bids increased for users that have a particular profile attribute. Conceptually, the bid increment indicates expectation of the ad being more effective when shown to or clicked by such users. Bid increments are already commonplace in display advertising platforms, where they are based on either demographic attributes, or broad, loosely defined categories [21]. In keyword advertising, the bid increment is charged if the advertising keyword matches those in user's profile. This notion integrates naturally with existing keyword campaigns in search and contextual advertising. The corresponding utility function is the total bid increment amount for clicks in matched settings. Note that this utility function underestimates the actual increase in revenue, as it does not account for revenue increases resulting from improved re-ordering of the ads due to bid increment. Because a higher bid can only increase the ad's position in a second-price auction, it makes corresponding clicks more likely, hence amplifying the gains. The utility formulation also does not account for expected gains in revenue due to cost-per-click increases for non-matched ads.

4.1 Utility Computation

Estimating the bid increment utility of a set of keywords requires computing the probability that, in the next time step, the user will click on an ad for one of those keywords or their graph neighbors. This is done by estimating the expected future clicks individually for each keyword involved, then combining these (according to the keyword graph) to compute overall utility.

We employ a machine learning approach for this estimation: a parameterized function is trained on historical data to minimize the difference between observed clicks and those predicted by the function, based on a set of features. Historical data comes from users who do not opt-out of server-side profiling (some number of such users can always be retained via incentives). The features are functions of the keyword, context and/or user that assist the model in predicting whether the user will click on an ad for the keyword in the future. Our model incorporates three feature sets:

- ***User Prior Features*** based on the counts of past searches and ad clicks for the user, which can be stored alongside the cache and profile client-side and incremented continuously;
- ***Recency Features*** based on recency of past occurrences for each considered profile/cache keyword, captured via 10 geometrically increasing lookback windows;
- ***Time-weighted Frequency Features*** based on heuristic time-decay functions that assume that the probability of a future click decreases with time, yet increases with the count of past occurrences.

Logistic regression based on the L-BFGS optimizer was chosen as the learning algorithm for utility prediction as it outperformed a number of other learners in pilot studies. Once the utility prediction function is trained, online profile construction is performed by considering every keyword (profile, cache, and context, including search queries and advertisements shown) and their children as candidates for inclusion in the profile, using predictions of their expected clicks in the subsequent time interval. Iteratively, the keyword with highest incremental utility is added to the profile, where incremental utility is the sum of the keyword's and its neighbors' expected clicks, subtracting those already covered by keywords selected in earlier iterations. Because the utility function is submodular and monotone, this algorithm is guaranteed to find a profile with utility that is at least $1 - {}^{1}\!/_{e} = 63\%$ of optimal.

5 Results

Experimental evaluation of the proposed approach for constructing compact keyword profiles was performed using advertising system logs from the Bing search engine over a two-week period. The first week's data was used for training of the utility predictor as described in Section 4. The training set contains the candidate keywords and their features, extracted over the week-long period, with labels (clicks) obtained from the subsequent one-day interval. Although this reduction of utility prediction to a standard supervised learning task neglects the online setting (i.e., the pool of candidates during online client-side construction is significantly smaller), it provides a reasonable batch-setting approximation, leaving truly online approaches as an interesting direction for future work. The experiments relied on the keyword graph used for matching related keywords in production.

With the utility predictor trained on past data, we evaluated the efficacy of online client-side profiling on a held-out set of over 20,000 users during the subsequent week. Profiles constructed in the server-side setting (using the complete user behavior history) were compared to those constructed in the online client-side setting (using a small cache of user behavior). Figure 2 illustrates the relative performance of client-side personalization with respect to server-side personalization for different profile sizes, demonstrating that even modest cache sizes provide performance that is comparable to server-side profiling with complete history. Indeed, for a profile size of 20 and a cache size of 40, online client-side profiling captures 98% of the revenue gain of server-side profiles, while giving full control of data and privacy to the user. Such settings are reasonable as they allow fitting both the profile and the cache with corresponding features into the cookie size limit of 4 kilobytes.

Figure 3 compares the performance of a sample client-side profiling setting to that of server-side profiling, and also to the maximum achievable performance. The latter corresponds to an oracle selecting the optimum profile from the user's complete history of past behavior to obtain maximum future utility presciently, thus bounding the amount of improvement that could be obtained with more sophisticated features or learners.

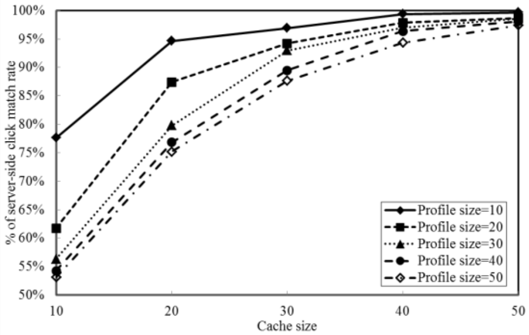


Figure 2: Client-side profiling accuracy (w.r.t. server-side) for different profile sizes

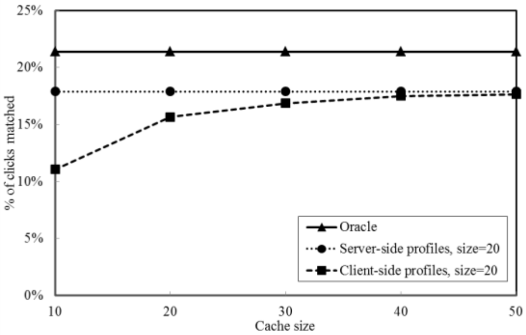


Figure 3: Comparison of client-side, server-side and maximum achievable accuracy

As these results demonstrate, our overall approach to constructing keyword profiles achieves a significant fraction of the maximum possible performance (83% of the oracle, for profiles of size 20). The oracle's upper bound of 22% of matched clicks captures the low overall predictability of future ad clicks. Finally, we note that if advertisers opted for a 25% bid increment (an average of comparable increments seen in existing targeting experiments), client-side keyword profiling would increase overall search engine revenue by over 6%, a sizeable gain that can be realized in a privacy-friendly way.

6 Related Work and Conclusions

Previous work on profile construction for ad personalization has focused on display advertising, where profiles consist of high-level categories also known as behavioral targeting segments [3]. As an alternative to predefined segments, Yan et al. [21] evaluated whether clustering methods to identify groups of users that show similar CTR behavior. Le et al. [12] also investigated clustering users based on their browsing behavior, observing that users who visit the same web pages have similar interests and thus click on similar ads.

Personalization for search advertising was previously considered only in the context of modifying clickthrough prediction estimates. Chen et al. [4] propose a latent-topic model for user-dependent CTR prediction, where each user is represented by a mixture of automatically derived topics. Similarly to other work on behavioral targeting to date, latent topics provide a relatively low targeting granularity, and are also not as transparent as keyword based profiles. Recent work on user-dependent CTR prediction for sponsored search by Cheng and Cantú-Paz [5] used two types of user features: demographic categories, and features derived from the user's historical CTR behavior for a given advertiser or query, presuming complete server-side history. Compared to that work, keyword profiles provide orthogonal benefits of making personalization explicit to advertisers via bid-increments, while allowing for user transparency and client-side storage. Combining prior work on personalizing CTR prediction with keyword profiles and deriving advertiser-side mechanisms for pricing increments are two interesting directions for future work.

In contrast to ad personalization, search result personalization has been a topic of active research for many years. Kelly and Teevan [11] provide a survey of techniques that build profiles of users based on their past behavior, using a variety of signals that include query history [16], browsing activity [13], or a combination of the two [1][18].

To our knowledge, this work is first to consider the problem of online construction of client-side keyword user profiles. Our framework allows making profiling transparent to users with little storage or communication overhead. Initial results demonstrate that maintaining client-side profiles incrementally through caching and greedy optimization enables ad platforms to allow users to opt-out of server-side logging without significant losses in revenue from personalization based on complete user history. While a number of interesting research challenges remain in developing better online learning algorithms for this problem, we believe our general approach has significant potential for improving personalized advertising.

References

[1] M. Bilenko, R. White, M. Richardson, G.C. Murray. Talking the talk vs. walking the walk: salience of information needs in querying vs. browsing. *SIGIR-2008*.

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

[3] Y. Chen, D. Pavlov, J. Canny. Large-scale behavioral targeting. *KDD-2009*.

[4] Y. Chen, M. Kapralov, D. Pavlov, J. Canny. Factor modeling for advertisement targeting. *NIPS-2009*.

[5] H. Cheng, E. Cantú-Paz. Personalized click prediction in sponsored search. *WSDM-2010*.

[6] M. Fredrikson and B. Lifshits. REPRIV: Re-Envisioning In-Browser Privacy. MSR Tech. Report, 2010.

[7] S. Guha *et al*. Serving Ads from localhost for Performance, Privacy, and Profit. *HotNets 2009*.

[8] S. Gupta, M. Bilenko, M. Richardson. Catching the drift: Learning broad matches from clickthrough data. *KDD 2009*.

[9] D. Heckerman, E. Horvitz and B. Middleton. An Approximate Nonmyopic Computation for Value of Information. *IEEE PAMI* 15: 292-298, 1993.

[10] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. *WWW 2006*.

[11] D. Kelly and J. Teevan. Implicit feedback for inferring user preference. *SIGIR Forum, 2003*.

[12] T. Li *et al*. A Markov chain model for integrating behavioral targeting into contextual advertising. *Ad-KDD 2009 Workshop*.

[13] B. Marlin. Modeling user rating profiles for collaborative filtering. *NIPS 2004*.

[14] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. *KDD 2009.*

[15] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming, 14:265-294*, 1978.

[16] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. *WWW 2006*.

[17] F. Radlinski *et al*. Optimizing relevance and revenue in ad search: a query substitution approach. *SIGIR 2008*.

[18] J. Teevan, S.T. Dumais, E. Horvitz. Personalizing search via automated analysis of interests and activities. *SIGIR 2005*.

[19] V. Toubiana *et al*. Adnostic: Privacy Preserving Targeted Advertising. *NDSS 2010*.

[20] Y. Xu, B. Zhang, Z. Chen, K. Wang. Privacy-Enhancing Personalized Web Search. *WWW 2007*.

[21] J. Yan *et al*. How much can behavioral targeting help online advertising? *WWW 2009*.