Predictive Client-side Profiles for Personalized Advertising

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

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

ABSTRACT

Personalization is ubiquitous in modern online applications as it provides significant improvements in user experience by adapting it to inferred user preferences. However, there are increasing concerns related to issues of privacy and control of the user data that is aggregated by online systems to power personalized experiences. These concerns are particularly significant for user profile aggregation in online advertising.

This paper describes a practical, learning-driven client-side personalization approach for keyword advertising platforms, an emerging application previously not addressed in literature. Our approach relies on storing user-specific information entirely within the user's control (in a browser cookie or browser local storage), thus allowing the user to view, edit or purge it at any time (e.g., via a dedicated webpage). We develop a principled, utility-based formulation for the problem of iteratively updating user profiles stored client-side, which relies on calibrated prediction of future user activity. While optimal profile construction is NP-hard for pay-per-click advertising with bid increments, it can be efficiently solved via a greedy approximation algorithm guaranteed to provide a near-optimal solution due to the fact that keyword profile utility is submodular: it exhibits the property of diminishing returns with increasing profile size.

We empirically evaluate client-side keyword profiles for keyword advertising on a large-scale dataset from a major search engine. Experiments demonstrate that predictive client-side personalization allows ad platforms to retain almost all of the revenue gains from personalization even if they give users the freedom to opt out of behavior tracking backed by server-side storage. Additionally, we show that advertisers can potentially increase their return on investment significantly by utilizing bid increments for keyword profiles in their ad campaigns.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; H.3.4 [**Systems and Software**]: User profiles and alert services.

General Terms

Algorithms, Human Factors, Experimentation, Measurement

Keywords

Online advertising, client-side personalization.

KDD'11, August 21–24, 2011, San Diego, California, USA.

1. INTRODUCTION

Personalization is a core component of many web applications, where its uses vary from re-ranking search engine results to recommending items in domains such as news or online shopping. Traditional uses of personalization center on customizing the output of an information system for a given user based on attributes composing their profile. Profile attributes may be explicitly or implicitly obtained, where explicit attributes are provided by the user or computed deterministically (e.g., user-submitted demographics or IP-based location). Implicit user attributes are inferred based on the logs of the user's prior behavior, e.g., past searching, browsing, reviews or shopping transactions. A wide variety of personalization approaches have been proposed in recent years; notable examples include algorithms that leverage preference correlations across users (i.e., collaborative filtering), and methods that use past behavior to assign users to one or more pre-defined categories (i.e., "targeting segments" in online advertising).

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 advertiser-defined categories for behavioral targeting in display advertising [3][37] and low-dimensional latent topics in collaborative filtering methods based on matrix decomposition [24]. 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 challenged by consumer and privacy advocates due to the fact that it limits users' ability to view and control data associated with their behavior, raising the need for privacy-enhanced personalization methods. In the context of personalized search, methods have been proposed for constructing category-based profiles on a user's machine (*client-side*), where they are employed to re-rank search results [34][36]. In online advertising, several alternative ad delivery architectures have been proposed based on client-side category-based profiles used to perform ad selection locally [13][10][35]. The previous approaches' reliance on broad categories for representing user interests is a significant barrier for their use in search and contextual advertising, where campaigns target highly specific intents via bids on individual keywords (short phrases that match the query or webpage exactly or approximately). Furthermore, these approaches require significant architectural changes to ad delivery pipelines and installation of additional components client-side both present significant barriers to adoption.

In this paper, we describe a novel approach to keyword-based personalization for search advertising that is practical, principled, and privacy-friendly. Based on *keyword bid increments* that allow differentiating between users with casual and long-term topical interests, the approach naturally integrates with existing ad delivery platforms, campaigns, and bid optimization frameworks, allowing advertisers to experiment with highly-granular ad personalization without significant infrastructure investments.

While highly practical, the described approach for keyword profile construction is derived from a principled utility-based framework. We demonstrate that bid-increment-based keyword profile utility is a submodular function: it has the intuitive property of diminishing returns with increasing profile size. Submodularity allows employing a simple yet highly effective approximate algorithm for profile construction. This enables real-time personalization based on client-side profile storage, avoiding server-side aggregation of user data, a major area of concern for consumer advocates.

The economic trade-offs of privacy-friendly policies are a key issue for the industry. For the proposed approach, we compare the performance of online, client-side profiling to full-history server-side profiling. Experiments on real-world large-scale datasets demonstrate that client-side profiling retains almost all revenue gains that server-side personalization yields, while allowing users to opt out of server-side logging and gain full control of their behavioral history.

Following are the paper's key contributions:

- Introducing the problem of keyword-based user profiling for online advertising, an emerging industry application that enables advertisers to customize existing search advertising campaigns based on users' prior behavior;
- Proposing a principled utility-based formulation for the client-side profile construction problem, applying it to the bid-increment setting for keyword advertising, and deriving an efficient algorithm for profile updates that does not require server-side storage of user data;
- Experimentally evaluating the trade-offs of enabling users to opt out from server-side storage of their behavioral history, and demonstrating that the corresponding "cost of privacy" for ad platforms performing personalization is minimal.

The rest of the paper is organized as follows. Section 2 describes a formal specification of the optimal profile construction problem in general, and its specialization to the client-side setting. Section 3 introduces keyword-based profiles for search advertising and validates their utility on a large real-world behavioral dataset. Section 4 describes the profile construction algorithm, while Section 5 explains the machine learning approach behind optimal profile construction. Experimental evaluation of the approach on real-world large-scale dataset from a major search engine is described in Section 6. Discussion of future work is provided Section 7. Section 8 provides an overview of related work and finishes with concluding remarks.

2. CONSTRUCTING USER PROFILES

To develop a principled foundation for deriving profile construction algorithms, we begin by formalizing the problem of profile construction for optimizing task-specific utility.

2.1 Profile Construction: Formal definition

Let $\mathcal{V}$ be the finite domain of items onto which observed user behavior is mapped, Φ be the domain of item descriptors, and $C = \mathcal{V} \times \Phi$ be the domain of observed contexts. For example, in search advertising, $\mathcal{V}$ is the set of all advertiser-bid keywords to which user queries are matched, and $\Phi = \mathbb{R}^d$ represents vectors of features associated with each query that has been mapped to the keyword. Then, every user query is represented as an observed context $c = (v, \phi)$ where $v \in \mathcal{V}$ is the most relevant ad keyword for the query, and $\phi \in \Phi$ is a vector of features that may be based on various attributes of the query and keyword, e.g., timestamp, similarity between q and v, user location, query or keyword category, etc.

Let $\mathcal{H}$ be the domain of all sequences of observed events corresponding to user behavior history, and $\mathcal{P}$ be the domain of profile representations. A *profile construction* function $f: \mathcal{H} \to \mathcal{P}$ maps a sequence of contexts observed over a time period up to t, $H = \left(c^{(t_1)}, .., c^{(t_n)}\right), t_i < t$, to a user profile $p \in \mathcal{P}$. The profile representation can be easily extended to include explicit profile attributes, e.g., demographic data.

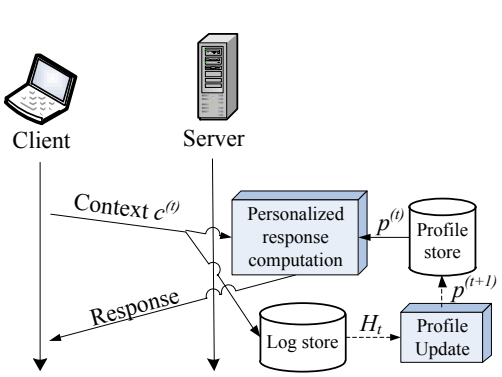
The objective of profile construction is to maximize a utility function that captures the increase in performance for the task at hand (representing the benefits of personalization). This utility function u can be computed post-hoc by evaluating performance of the profile constructed during a preceding time interval, $u: \mathcal{H} \times \mathcal{P} \to \mathbb{R}$. Then, the optimal profile construction function f^* is one that maximizes utility:

$$f^* = \underset{f}{\operatorname{argmax}}\, \mathbb{E}\left[u\left(f(H_t), c^{(t+1)}\right)\right], \qquad (1)$$

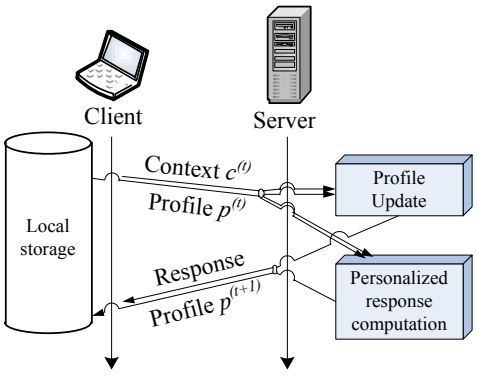
where the expectation is computed across the distribution of contexts subsequent to profile construction period.

2.2 Client-side Profile Construction

Motivated by the significant privacy concerns with server-side storage of user behavior history, we consider storing the user data employed for profile construction only on the user's machine (client-side), with the platform providing the user with a way to inspect, edit or delete it. The profile is sent to the server at interaction time along with the user request and current context (e.g., query or webpage id), updated server-side and returned to user together with the personalized response (which may be computed simultaneously or sequentially with profile update, depending on latency requirements and computational costs). The key distinction from the user privacy perspective is that no information associated with the user is retained server-side after response computation and profile update. Figure 1 illustrates the distinction between traditional server-side profiling and the client-side approach. Client-side storage of profiles is supported by current web browsers natively via cookies. While in the near future several additional mechanisms (e.g., HTML5 local storage) will provide increased client-side storage capacity, communication costs provide strong motivation for maintaining a limited-size profile.



(a) Server-side profiles: The server stores user history and profiles, updating them synchronously or asynchronously.



(b) Client-side profiles: The client stores the profile, while the server only performs the profile update.

Figure 1: Comparison of personalization with server- and client-side profiles.

Because client-side profiles are sent to the platform for their utilization in personalizing the user experience, this mechanism provides privacy under the assumption of compliance on behalf of the platform. However, we believe that this assumption is realistic: ad platforms already provide privacy-related guarantees to users (e.g., limiting third-party dissemination of user data and its retention over certain time limits), and both regulatory and competitive pressures strongly incentivize compliance. We note that alternative mechanisms that attempt to provide privacy guarantees by performing personalization and profile maintenance client-side [13][10][35] require users to install additional browser components, and assume significant modifications in ad platforms' delivery and reporting infrastructure, which presents a significant challenge to their implementation.

3. PROFILES FOR CPC ADVERTISING

3.1 Bid Increments and Keyword Profiles

Unlike display advertising platforms where inventory is sold on a pay-per-impression basis (a.k.a. CPI or CPM), search and contextual ad platforms require advertisers to submit bids on individual keywords which are matched against queries or webpage content (either exactly or approximately). If selected to be shown, advertisers are charged only when a click is observed (a.k.a. CPC), with cost assessed by discounting the bid via a second-price auction mechanism [8].

The ad inventory format impacts the choice of user profile representation that is exposed to advertisers: e.g., in display advertising where impressions are sold in large blocks, high-granularity profiles are used that are composed of such features as demographic attributes (e.g., "*Young urban males*") or broad categories (e.g., "*Car enthusiasts"*).

In search and contextual ad platforms, keyword bids are inherently highly specific and can already be combined with demographic and location attributes. Hence, it is not surprising that advertisers have expressed an interest in the ability to specify keyword-specific *bid increments* by which the keyword bid is increased for users who have shown a historical interest in the keyword's narrow topic. This is the core of the emerging application setting that we address in this paper: constructing user profiles comprised of bidded keywords that trigger advertiser-specified bid increments when the user's context (query or webpage) matches the keyword in the profile (exactly or approximately).

By allowing advertisers to target keyword profiles, pay-per-click campaigns can be refined to increase bids for those users for whom they are likely to be more effective. For example, a diving equipment store may be willing to pay more for diving-keyword-related clicks of users predicted to have a long-term interest in diving, since they are more likely to purchase high-end items. We also note that bid increments are currently commonplace in display advertising, however there they are based only on explicitly known demographic attributes, or broad, loosely defined categories.

Another motivation for advertisers to employ bid increments is exploiting the variability of users' conversion rates depending on whether the keyword is in their profile or not: the following section provides analysis demonstrating the presence of this effect in real-world data.

3.2 Conversion Analysis for Profiling

To verify that keyword profiles constructed based on past searching behavior provide value to advertisers that justifies setting bid increments, we compared conversion rates for keywords that were seen in a user's past behavior, versus that for keywords that were previously not seen. Conversions are post-click events reported by advertisers that signal a user action, such as a purchase, a reservation, or answering a questionnaire, and conversion rate is defined as the fraction of ad clicks that have led to a conversion.

We analyzed 2 months of search advertising logs for a sample of 2.4 million users of Bing search engine (randomly selected from US-based active users logged in with a persistent ID over the 2-month period), and have measured the distribution of conversion rates vs. the time since the *preceding* occurrence of the keyword in the user's prior query and click behavior. Figure 2 illustrates the relative differences in conversion rate for keywords vs. time since last occurrence, where the difference is computed w.r.t. to the prior conversion probability (normalized for variability across keywords).

These results demonstrate that there is a very significant effect that previous observations of the keyword have on

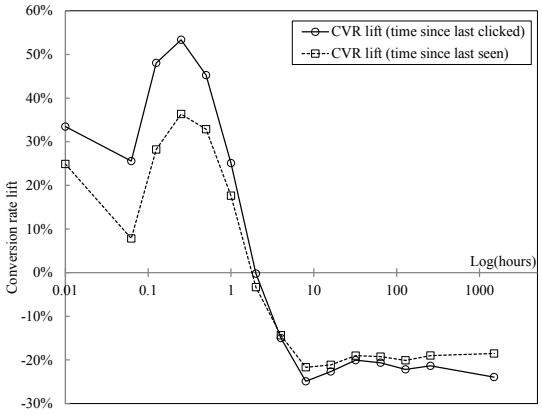


Figure 2. Conversion rate lift as a function of time since last observation of keyword in query or click.

conversion rates: overall, conversions are more likely in the initial several hours since the previous occurrence of the keyword. Average conversion rates decline significantly after several hours, possibly representing keywords that are regular repeating queries for users (e.g., daily weather or stock ticker queries), which are arguably less likely to be converted when their ads are clicked. However, conversion rates for individual keywords show a significant variety in temporal dynamics, with many keywords showing increased conversion lift over multi-day periods.

Although conversion events represent a variety of user actions across different advertisers and hence their specific value can vary between campaigns, it is universally accepted that increases in conversion rates are desirable to advertisers, as they indicate higher user engagement for paid clicks. Thus, the results above indicate that advertisers have a strong incentive to differentiate their bids for users who have exhibited prior activity for the campaign keywords. Interestingly, this also indicates that *negative* bid increments may be of value to advertisers as well, allowing them to lower bids for users whose past behavior indicates lower conversion propensity. We note that, given advertisers' bid increment amounts and known conversion curves such as those above, we can threshold inclusion of keywords by predicted conversion lift to reflect advertisers' expected ROI.

4. PROFILE CONSTRUCTION

4.1 Utility of Keyword-based Profiles

To design a profile construction policy that is optimal w.r.t. expected utility as defined by Eq.(1), we must formulate the personalization utility provided via the bid increment mechanism in pay-per-click advertising. From ad platform's perspective, personalization utility is equivalent to the gain in revenue that can be attributed to profiles. Profiles can be used in several stages of ad selection within the advertising platform, e.g., retrieval of candidate ads, CTR estimation, and ad ranking; we discuss these possibilities in Section 7 in more detail. For the present discussion, we focus on the scenario where profiles' only use is to trigger bid increments, noting that leaving other profile uses out of the analysis and experimental results, we only underestimate the potential revenue increases, and hence our results provide a lower bound on profile utility.

Then, the utility of a keyword profile $p^{(t)}$, composed of n phrases $k_1,..,k_n$, can be formulated as the revenue gain collected from future clicks that match profile components. The expectation of this gain at profile construction time t for the next ad-serving context $c^{(t+1)}$ (subsequent to profile construction) is then:

$$u(p^{(t)}) = \mathbb{E}[R(p^{(t)}, c^{(t+1)}) - R(c^{(t+1)})]$$

where the expectation is computed over the distribution of subsequent contexts $c^{(t+1)}$; $R(p^{(t)}, c^{(t+1)})$ and $R(c^{(t+1)})$ represent revenue earned in the subsequent context with the profile and without, respectively. For CPC advertising with bid increments, this expectation is equal to:

$$u(p^{(t)}) = \sum_{c^{(t+1)} \sim p^{(t)}} \Pr(c^{(t+1)}) \sum_{a \in A(c^{(t+1)})} \Pr_{click}(a)\, \Delta Bid(a, p^{(t)}) \quad (2)$$

where we utilize the fact that the expectation needs to be computed only over contexts (queries) that match the keywords comprising the profile (denoted as $c^{(t+1)} \sim p^{(t)}$) and hence can be enumerated, with $\Pr(c^{(t+1)})$ representing the probability of each such context, $A(c^{(t+1)})$ being the potential ads for the context, $\Pr_{click}(a)$ denoting the probability that the ad will be displayed and clicked, and $\Delta Bid(a, p^{(t)})$ is the bid increment for ad *a* with the given profile (which may be zero if the ad does not match the profile).

Note that this definition does not take into account auction pricing effects that are two-fold due to discounting of the bids in second-price auctions: on one hand, higher bids for incremented ads will increase the effective costs-per-click for non-incremented ads, while on the other hand, the increases in actual charged increments may be less than the full bid increments. Because these second-order effects are opposing, we consider their overall effect marginal in comparison to the first-order effects on which we focus.

4.2 Submodular Profile Construction

We note from Figure 2 that keywords observed in a recent time window exhibit a higher than expected conversion rate. This suggests maintaining a *cache* of queries that is stored client-side along with the profile, and used to enhance the pool of candidate profile keywords in addition to previous profile and current context. The size of such a cache would be determined by storage and communication limitations. We note that the cache keywords are distinct from those in the profile (though some could be in both): the cache retains keywords solely for enhancing the profile construction candidate pool, while profile keywords are selected to maximize predicted utility in Eq.(2), and may be chosen from either the cache, the previous profile or the present context. Hence, profile $p^{(t)}$ transmitted from the client in Fig.1(b) contains both actual profile keywords used for ad targeting, and cache keywords used for profile updating.

To obtain a keyword selection algorithm that optimizes the utility of Eq.(2), it must be decomposed into contributions of each individual keyword comprising the profile. However, because of approximate matching, two different keywords k_i and k_j, may share the ad inventory: $A(k_i) \cap A(k_j) \neq \emptyset$, which implies that it is not possible to decompose the utility in Eq.(2) across candidate keywords. E.g., an ad that with an approximate-match bid on keyword "*jumbo mortgage*" should also be matched to the query "*jumbo interest rates*".

Such approximate matching is represented by a directed keyword graph with vertices corresponding to bidded keywords in the platform's inventory, and edges connecting keywords for which ads are approximately matched. Figure 3 below illustrates the necessity of the graph being directed: while more restrictive keywords (e.g., "*seattle mortgage*") match ads with bids for less restrictive keywords (e.g., "*home loans*"), the opposite is not true (since advertisers bidding on "*seattle mortgage*" are interested only in a small subset of all users looking for "*home loans*").

Building keyword graphs is a well-studied problem as they are a standard component for keyword ad platforms. In this work we use a large production graph with tens of millions of keyword nodes that is produced using an ensemble of data mining and machine learning algorithms that identify related keywords based on a variety of sources (query reformulations in user sessions, ad campaign co-occurrence, etc.), and rank them using a combination of supervised data and click-through feedback [15][16][31].

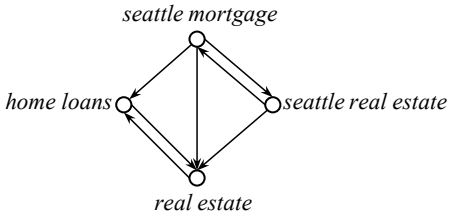


Figure 3: Sample fragment of the keyword graph

Because of the overlap in ad inventory between keywords due to approximate matching, the profile utility function in Eq.(2) is *submodular*: when a keyword k is considered for inclusion in a profile, the utility gain from including it cannot be greater than that for any subset of the profile:

$$u\left(p^{(t)} \cup k\right) - u\left(p^{(t)}\right) \leq u\left(p^{(t)}_{subset} \cup k\right) - u\left(p^{(t)}_{subset}\right)$$

for $\forall k,\ \forall p^{(t)}_{subset} \subseteq p^{(t)}$. Informally, submodularity captures the diminishing returns effect of growing the profile: as more keywords are added, they match increasingly less new ads that are not already matched by keywords selected earlier.

Limits on profile size entail that the particular submodular optimization problem to which profile construction can be reduce is *budgeted maximum coverage* [18], which is NP-hard. However, it can be solved approximately using the greedy algorithm of Nemhauser et al. [15], which is guaranteed to produce a solution that is $\left(1 - 1/e\right) \approx 63\%$ optimal. The algorithm proceeds by starting with an empty profile and iteratively adding the keyword with maximum incremental utility gain $\Delta u\left(k, p^{(t)}\right) = u\left(p^{(t)} \cup k\right) - u(p^{(t)})$, which is:

$$\Delta u(k, p^{(t)}) = \sum_{c^{(t+1)} \sim k} \Pr\left(c^{(t+1)}\right) \sum_{\substack{a \in A\left(c^{(t+1)}\right): \\ \Delta Bid\left(a, p^{(t)}\right) = 0}} \Pr_{click}(a)\, \Delta Bid(a, k) \quad (3)$$

This expression captures the expected revenue gain due to keyword k being added to profile $p^{(t)}$. The gain is aggregated over all approximately matched contexts, $c^{(t+1)} \sim k$, corresponding to keywords in the graph that have edges connecting them to k. For these contexts, the expected gain is aggregated over the ads $A\left(c^{(t+1)}\right)$ that are matched to the context, excluding those that are already incremented due to them being approximately matched to a keyword previously in $p^{(t)}$. Thus, the greedy approximate profile construction algorithm maximizes profile utility in Eq.(2) by iteratively adding keywords with maximum incremental utility predicted by Eq.(3). Next section describes a machine learning approach for predicting this utility.

Finally, we note that while the naïve implementation of the Nemhauser algorithm is $O(nk)$, where k is profile size and n is the candidate pool of keywords, it can be sped up dramatically by using lazy evaluation [21], and empirically can be run for hundreds to thousands of keywords within the real-time constraints of modern ad delivery platforms.

5. PROFILE UTILITY ESTIMATION

Computing incremental keyword utility given by Eq.(3) requires predicting the probability that a given context matching a given keyword will occur in the future, as well as the probability that an ad matching that context will be clicked. We employ a machine learning approach for these prediction tasks: a parameterized function is trained on past user behavior data to optimize utility estimation using observed ad impressions and clicks. Relying on past data assumes that some fraction of users choose to not opt out from server-side profiling and continue having their behavior history logged. In light of historically low opt-out rates for privacy-preserving settings in various web applications and toolbars, this assumption is reasonable.

Model training proceeds by simulating the profile construction process at an intermediate time, with the set of candidate keywords collected over the preceding time interval yielding the set of training instances. True labels (actual utilities) are obtained based on the presence of the keyword candidate in subsequent behavior. The utility is 0 if the keyword was not matched to subsequent search contexts, or if it was matched and there were no ad clicks. If the keyword was matched to a future query leading to a corresponding ad impression, and a click was observed, utility is the corresponding bid increment value. The learning algorithm then attempts to identify the predictor parameters that minimize expected error (typically, by minimizing training set error under some regularization technique that prevents overfitting).

We experimented with three different learning algorithms: max-margin averaged perceptron [6], logistic regression trained using the L-BFGS algorithm with L1 and L2 regularization [1], and boosted decision trees [11]. Section 6.3 provides a summary of results for the different learners.

5.1 Predictive Features

Features are functions of the keyword candidates considered for inclusion in the profile and/or user properties that are informative for predicting the expected utility of a keyword. We experimented with three broad feature sets that capture key signals related to the prediction task at hand. Features are computed based on known user behavior (the current user's profile information, plus historical data from other users who have not opted out of server side logging), as well as user-independent keyword properties and historical statistics.

5.1.1 User Features

A user that has searched and clicked frequently in the past is more likely to search and click in the future. Thus, we gener-

ate three features: the number of times the user queried the search engine, the number of times the user clicked on an ad, and the fraction of searches that led to clicks. The number of clicks by the user yields the most directly informative signal, but it has high variance since ad-click events are relatively rare. Thus, the number of past searches is included to provide the learner additional information it can exploit to produce a better estimate of the user's propensity to click on ads. Storing the values needed to compute and update this feature takes only a few bytes, and hence can be trivially augmented to the client-side profile.

5.1.2 Keyword Features

Some keywords are more likely to be searched for, or have higher clickthrough rates than others. Based on historical data, we compute the probability that an average user will search or click ads for a given keyword, and utilize features derived from such probabilities. Because these features are user-independent, they can be stored server-side.

5.1.3 User-Keyword Features

A user's history of past behavior in the context of a candidate keyword provides strong clues about the likelihood of its appearance in the future search and/or click behavior. These features must be stored alongside the candidate keywords in the profile, however the occurrence history for each keyword can be compressed down to just several bytes by using counts on discretized, equal-sized time windows as well as a single value indicating how recently the counts were last "moved" from one time window to older time windows. Given this discretized history, for each candidate keyword features are generated representing the number of past searches, ad impressions and clicks for a set of geometrically increasing time ranges, both as total counts for each range, as well as the differences between adjacent ranges.

An additional set of features is derived from the same past occurrence data by applying several time-decay functions – linear and logarithmic – to aggregate the counts, weighing more recent occurrences higher. These features attempt to model the fact that that predictive value of past occurrence counts decays with time yet increases with more clicks on the same keyword.

6. EXPERIMENTAL RESULTS

6.1 Methodology

The proposed approach for constructing client-side keyword profiles was evaluated using two months of search and advertising behavior logs for 2.4 million users of the Bing search engine, sampled randomly from the overall, larger pool of bot-filtered US-based active users, where active users were defined as those users who had used the search engine (issued at least one query) on at least 30 of the 60 days in the time period. The first six weeks of data was used for training the utility predictor as described in Section 5 by simulating the profile construction process and using the subsequent behavior to obtain a training label (realized utility) for every keyword that was a candidate for inclusion in the profile. With the utility predictor trained on the first six weeks, the efficacy of online profiling was evaluated by simulating profile construction over the seventh week, using behavior following the construction period to estimate utility. Section 6.2 reports results using logistic regression, trained using the L-BFGS algorithm with a combination of L1 and L2 regularization, while section 6.3 provides a comparison of different learners. Experiments utilized the actual keyword graph used by Bing's advertising system for approximate matching of related keywords.

Recall that client-side profiles contain two portions: the profile, which matches keywords and triggers bid increments, and the additional keyword cache, used to enhance the profile construction candidate pool (see Section 4.2). Whereas the profile construction uses the utility model as outlined earlier, the cache is more straightforward so we chose least-recently-used caching, a standard caching approach that typically has good performance and is efficient to compute.

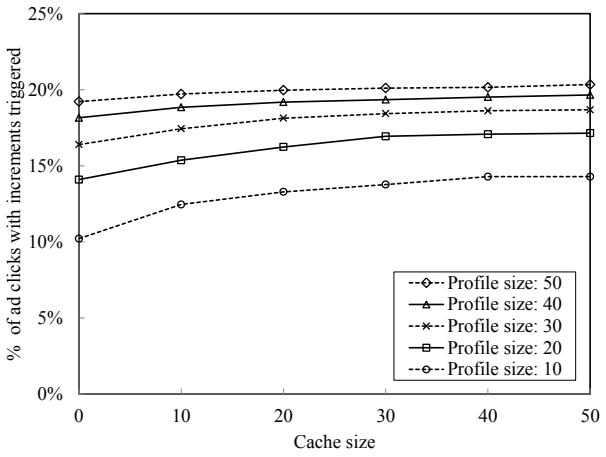
Because bid increments are not currently publicly available to all advertisers, we are unable to report utility in terms of actual revenue gains. However, it is possible to report the fraction of ad clicks in post-profile-construction behavior for which the profile matched the bidded keyword, and hence would have triggered the bid increment. This metric, percentage of incremented clicks, can be viewed as the percentage of ad revenue that would be increased via increments.

6.2 Client-side vs. Server-side Profiling

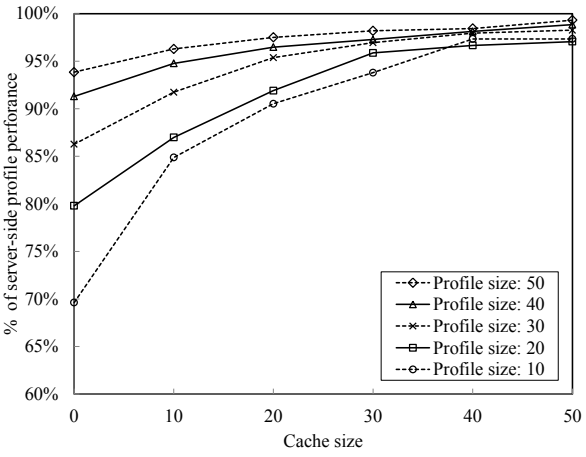
Figure 4(a) presents the performance of client-side profiling for a variety of profile and additional cache sizes. All differences between profile sizes for a given cache size are statistically significant ($p < .05$). Differences in cache size, for a given profile size, are also statistically significant up to and including cache size 30 ($p < .05$).

We were able to store more than 100 keywords, including historical information necessary to compute these features, within the size limitations of Web browser cookies (4KB), with space remaining for other site-dependent uses. Since the size of the recent query cache at least partially trades off with the profile size, we evaluate the effect of the amount of available query cache on utility, comparing it to having complete past behavior history (server-side profiles) for different profile sizes. Figure 4(b) illustrates this relative performance of client-side profiles (with their limited knowledge of user history) with respect to server-side profiles for different profile sizes. The figure demonstrates that maintaining a modest cache size alongside the profiles allows achieving performance comparable to that of server-side profiling, but without the need for storing the complete user behavior history. For example, if profiles are limited to 20 keywords, utilizing a cache of the 50 most recent queries allows capturing 97% of the revenue gain achieved by server-side profiles while providing users full control of their data. Given this, these results demonstrate the practicality of allowing users to opt-out from server-side profiling with minimal revenue or performance cost.

Figure 5 compares the performance of a 20-keyword client-side profile to that of server-side profiling along with upper and lower bounds on performance. Upper bounds are represented by the server-side and client-side "oracles", which select the candidate keywords presciently (with knowledge of actual future user behavior), thus always selecting keywords that actually receive future ad clicks, when possible. The oracle is not able to match 100% of future clicks because keywords it may select are limited to candidates derived from the user's known history, mirroring the setting for the



(a) Client-side profile utility (percentage of future clicks matching the profile).



(b) Relative client-side profile utility (utility as a proportion of server-side utility for profiles of the same size).

Figure 4: Client-side profile utility as a function of profile size and cache size.

actual profile construction algorithm. The difference between the client-side and server-side oracles is that the server-side oracle has access to the user's entire history, whereas the client-side oracle has access only to the keywords contained in the user's client-side profile and cache.

The oracle results provide a bound on the overall predictability of future ad click keywords from past behavior, showing the maximum potential improvement that could be obtained with more sophisticated features or learners. Note that the client-side oracle asymptotically converges to the performance of the server-side oracle as the cache size is increased, just as the performance of actual client-side profiles converges to that of server-side profiles.

Figure 5 also demonstrates that our approach to profile construction, including the scoring of keywords using machine learning and selecting the profile using submodular optimization, is able to outperform a simpler method such as using least-recently-used caching to construct the profile ("Cache as profile" in the figure). As the available pool of candidate keywords grows, the learned profile construction algorithm is able to leverage its knowledge to select those that are most likely to appear in the future, which may not even include the most recently seen keyword.

Overall, these results demonstrate that the proposed approach to constructing keyword profiles achieves a significant fraction of the maximum possible performance (e.g., 81% of oracle utility and 97% of non-oracle server-side utility for profiles of size 20). Concretely, our results imply that if advertisers opt for a 25% bid increment (an average increment seen in initial trials for select advertisers), keyword profile based personalization increases overall search engine revenue by over 4%, a sizeable gain that can be realized in a privacy-friendly way.

6.3 Learning Algorithms

To evaluate the impact of learning algorithm choice on performance, we have conducted experiments with three state-of-the-art learning algorithms: MART boosted trees, L-BFGS logistic regression, and max-margin averaged perceptron. Table 1 summarizes results obtained for 20-keyword profiles with a 50-keyword cache. Differences are statistically significant ($p < .05$).

Table 1. Impact of learning algorithm on performance.

Learning algorithm	Utility (% of future clicks with profile-triggered increments)
Boosted Decision Trees	16.63
Logistic Regression	16.86
MM-avgPerceptron	15.06

While all learners exhibit overall good results, averaged perceptron performs slightly worse. This may be due to the very significant class imbalance and overall noisy nature of the learning task at hand: because only a small fraction of keyword candidates lead to future clicks, online algorithms that optimize hinge loss (i.e., max-margin perceptron) may be less appropriate than those that optimize log-loss or exp-loss (i.e., logistic regression and boosting).

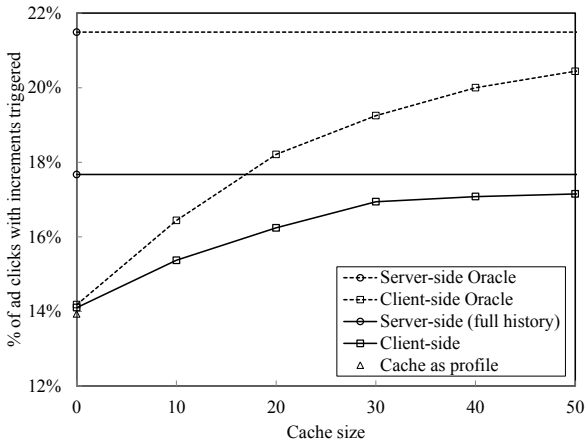


Figure 5: Comparison of client-, server-side and oracle utility (profile size 20).

7. DISCUSSION AND FUTURE WORK

While experimental results demonstrate that client-side profiles can be almost as effective as server-side profiles in capturing users' ad-related interests from prior behavior, we have only considered their utility in the context of their use for allowing advertisers to specify bid increments to vary bids for users of interest. However, there are a number of additional search advertising platform components where keyword-based profiles can be utilized, which present very interesting opportunities for future work:

- Ad selection: how can profile keywords be used to expand the set of advertisements selected for the auction in addition to those selected by just the context alone?
- CTR prediction: what features can be computed using profile keywords to enhance the accuracy of clickthrough prediction for advertisements? Are high-granularity profiles more or less informative than previously proposed low-granularity profiles composed of demographic and topical attributes [4][5]?
- Relevance filtering: can user profiles aid in computing ad copy and landing page relevance estimates used for filtering and ranking the selected ad candidates?
- Ranking: what are incentive-compatible auction mechanisms that can utilize information provided by user profiles for advertisers, e.g., by incrementing or reducing bids automatically to reflect adjustments in advertiser value inferred from profile information [26][12]?

The technique employed by the current paper centers on developing a principled algorithm for profile construction to optimize utility accumulated via bid increments. A sound approach to each of the above scenarios requires analogous derivation of profile utility for each task, and design of profile construction algorithms that attempt to maximize such utility, which presents an exciting array of opportunities for future work.

Finally, we note that utilizing additional signal sources, such as richer interaction data, has been shown to improve the accuracy of search intent inference [14], and may yield additional accuracy gains in the context of prediction tasks considered in this paper.

8. RELATED WORK AND CONCLUSIONS

Privacy-preserving personalization has received significant interest for a number of years from different areas of computer science; e.g., see surveys by Riedl [32] and Kobsa [19]. Advertising-specific issues at the intersection of privacy and personalization have not been considered until recently. Several architectures of personalization platforms coupled with delivery mechanisms have been designed with the goal of providing privacy guarantees [13][10][35], however, they require installation of additional client software and changes to the existing ad delivery and pricing mechanisms, and hence are not yet practical for the industry. The present paper takes a different approach and, instead of relying on theoretical privacy guarantees, provides an industry-ready method for enabling users to opt out from server-side tracking, under the assumption that advertising platforms that provide opt-out capability can be trusted, as they are legally required to comply with user agreements. Because deployment on a large, real-world ad platform was the primary motivation for this work, we believe that enabling the platform to serve personalized advertising to users who opted out of server-side logging is a key benefit of the proposed solution that makes it economically attractive.

Most previous work on personalization for advertising has focused on display advertising settings, where low-granularity profiles are employed to assign users to behavioral targeting segments: for example, Chen et al. apply a linear Poisson regression model to assign users to 450 pre-defined categories [3]. Yan et al. evaluate whether behavioral targeting provides value by clustering users into 20-160 segments and demonstrating that users in the same segment show similar CTR behavior [37]. Yan et al. also employs clustering to group users into segments based on their browsing history, assuming that users who visit the same web pages have similar interests, and thus will have an interest in similar ads [22]. In [29], Provost et al. consider a different avenue for privacy-friendly targeted display advertising: propagating user interests across inferred social networks.

As far as we are aware, only two papers considered personalization for search advertising, and both have focused on improving CTR estimates. Chen et al. [4] proposed a latent-topic model for user-dependent CTR prediction, where each user is represented by a mixture of automatically derived topics. Cheng and Cantú-Paz [5] have relied on inferred demographic features and users' historical clickthrough statistics across advertisers. In contrast, the present paper focuses on deriving a personalization approach that empowers advertisers to utilize user profiles in their campaigns, while enabling users to take control of their data. As mentioned in Section 7, there exists a number of interesting directions for hybridizing this previous work with the present paper, such as evaluating the use of keyword-based profiles for CTR prediction.

While employing user profiles in advertising is a nascent application, search result personalization has been a topic of active research for many years. Kelly and Teevan [17] provide a survey of techniques that construct profiles of users based on their past behavior. Such techniques typically rely on query history [30][33], browsing activity [24], or a combination of these [1][2][22][25][34]; see also the study of Dou et al. that compares and evaluates different search personalization strategies [7]. Because relevance improvements are the primary target of personalization in search, the corresponding utility functions differ significantly from advertising, where personalization utility is revenue-driven.

To our knowledge, this paper is first to address online construction of keyword user profiles for online advertising, as well as to consider client-side ad personalization that does not involve installing additional software or significantly changing the ad delivery infrastructure. We hope the proposed approach has significant potential not only for the personalization of advertising, but also for other information retrieval systems, such as search engines.

9. REFERENCES

[1] G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. *ICML 2007*.

[2] 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*.

[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. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. *EMNLP 2002*.

[7] Z. Dou, R. Song, J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. *WWW 2007*.

[8] B. Edelman, M. Ostrovsky, M. Schwarz. Internet advertising and the generalized second-price auction: selling billions of dollars worth of keywords. *American Economic Review* 97(1):242-259, 2007.

[9] D. Fain and J. Pedersen. Sponsored search: A brief history. In *Second Workshop on Sponsored Search Auctions*, 2006.

[10] M. Fredrikson and B. Livshits. RePriv: re-envisioning in-browser privacy. *IEEE Symposium on Security and Privacy 2011*.

[11] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29(5): 1189-1232, 2001.

[12] A. Goel. Hybrid keyword search auctions. *WWW 2009*.

[13] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis. Serving ads from localhost for performance, privacy, and profit. *HotNets 2009*.

[14] Q. Guo and E. Agichtein. Ready to buy or just browsing? Detecting web searcher goals from interaction data. *SIGIR 2010*.

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

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

[17] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum, 2003*.

[18] S. Khuller, A. Moss, and J. Noar. The budgeted maximum coverage problem. *Information Processing Letters* 70(1):39-45, 1999

[19] A. Kobsa. Privacy-enhanced personalization. *Communications of ACM* 50(8), 2007.

[20] A. Krause and E. Horvitz. A utility-theoretic approach to privacy and personalization. *AAAI 2008*.

[21] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. *KDD 2007*.

[22] L. Li, Z. Yang, B. Wang, and M. Kitsuregawa. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. *ADWM 2007*.

[23] T. Li, N Liu, J. Yan, G. Wang, F. Bai, Z. Chen. A Markov chain model for integrating behavioral targeting into contextual advertising. *AdKDD 2009 Workshop*.

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

[25] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. *WSDM 2011*.

[26] P. Milgrom and R. Weber. A Theory of auctions and competitive bidding. *Econometrica* 50:1089-1122, 1982.

[27] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. *SIGIR 1994*.

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

[29] F. J. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray. Audience selection for on-line brand advertising: privacy-friendly social network targeting. *KDD 2009*.

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

[31] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, L. Riedel. Optimizing relevance and revenue in ad Search: a query substitution approach. *SIGIR 2008*.

[32] J. Riedl. Personalization and Privacy. *IEEE Internet Computing* 5(6): 29-31, 2001.

[33] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any efoort from users. *WWW 2004*.

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

[35] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, S. Barocas. Adnostic: privacy preserving targeted advertising. *NDSS 2010*.

[36] Y. Xu, B. Zhang, Z. Chen, K. Wang. Privacy-enhancing personalized web search. *WWW 2007*.

[37] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, Zh. Chen. How much can behavioral targeting help online advertising? *WWW 2009*.