Computer Networks 31 (1999) 1259–1272

COMPUTER NETWORKS

Unintrusive customization techniques for Web advertising

Marc Langheinrich [*,1], Atsuyoshi Nakamura [1], Naoki Abe [1], Tomonari Kamba [1], Yoshiyuki Koseki [1]

NEC Corporation, C&C Media Research Laboratories, 1-1-4 Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 216-8555, Japan

Abstract

Most online advertisement systems in place today use the concept of consumer targeting: each user is identified and, according to his or her system setup, browsing habits and available off-line information, categorized in order to customize the advertisements for highest user responsiveness. This constant monitoring of a user's online habits, together with the trend to centralize this data and link it with other databases, continuously nurtures fears about the growing lack of privacy in a networked society. In this paper, we propose a novel technique of adapting online advertisement to a user's short term interests in a non-intrusive way. As a proof-of-concept we implemented a dynamic advertisement selection system able to deliver customized advertisements to users of an online search service or Web directory. No user-specific data elements are collected or stored at any time. Initial experiments indicate that the system is able to improve the average click-through rate substantially compared to random selection methods. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Online advertisement; World-Wide Web; Personalization; Privacy; Electronic commerce

1. Introduction

The World-Wide Web (WWW) continues to grow at an astonishing rate [17]. For the foreseeable future, advertisement remains the single major source of revenue for most companies on the Web since many users are not yet willing to pay for online services such as search engines, Web directories or online magazines.

In order for Web advertisement to be effective, advertisers increasingly rely on targeting techniques that invade a user's privacy. Some of the largest commercial sites on the World Wide Web recently agreed to feed information about their customers' reading, shopping and entertainment habits into a central system, mostly without the user's knowledge [9].

Surveys [20] indicate that users are beginning to value their online privacy more and more. Widespread consumer protest last year prompted online giant AOL to reverse its plans of selling information from its customer database to an online marketing firm [25].

Instead of amassing more and more information about each user, we propose a less intrusive approach: a low-data oriented customization sufficient to capture the short-term interests of users of Web directory and search services. As a first step, we have implemented an advertisement server system for short-term advertisement customization in order to measure the level of adaptivity that is possible with minimal data.

Our system relies only on search keywords supplied by a user to a search engine. Based on the user's current interests (as expressed by the cho-

* Corresponding author.
[1] E-mail: {marc,atsu,abe,kamba,koseki}@ccm.cl.nec.co.jp

1389-1286/99/$ – see front matter © 1999 Published by Elsevier Science B.V. All rights reserved.
PII: S1389-1286(99)00033-X

VIANT-1050

sen search keyword) the system dynamically selects a best matching advertisement. By only relying on one or more keywords, no user-specific data is collected or retained, allowing us to provide customized advertisements without invading user privacy. Alternatively, in a browsable directory setting, our system observes the URL of the page the user is requesting and selects the most appropriate advertisement for this page.

The following sections will first describe current techniques for advertisement selection, then contrast this with our plans for an unintrusive advertisement system called *ADWIZ*. After giving some implementation details of our first prototype we will summarize some simple, preliminary experiments we conducted and close with comments on future and related work.

2. Online advertisement

Advertisement is still the single most important revenue for many companies on the Web. After subscription-based models failed to catch on with subscribers and with digital payment standards yet to emerge, the often criticized *banner* ad continues to dominate spending in online advertising [24]. There is seemingly no end in sight: according to marketing research companies, revenue figures from online ad sales continue to grow at a rate of more than 200% a year [10], with expected revenues of over 2 billion dollars in 1998 alone [11].

2.1. Technical background

The basic concept of banner advertising is the display of a rectangular image (see Fig. 1) close to the top of the Web page. Clicking on the image (often encouraged with explicit "Click here for more information" text embedded in or next to it) will take the user to a new page, presumably on the advertiser's Web site, where detailed product descriptions or order information can be found.

Fig. 1. A typical advertisement *banner* on the Web. Most banner advertisements on the Web today are GIF or JPEG files in a rectangular format, suitable for display at the top or bottom of a Web page. However, many advertisement systems also support both smaller sizes (for example for displaying them on a navigation sidebar) and other media formats (such as HTML tables or even Java programs).

The technical details of both HTML [18] (the markup language used to describe Web page content) and HTTP [8] (the protocol used to request and transmit Web pages) make it possible to separate page content and advertisement, thus enabling dynamic advertisement selection every time a Web page is requested.

The basic process is outlined in Fig. 2: the Web page of an online service (the 'publisher') contains a link to a banner advertisement. Although the content of the original Web page (step 2) stays the same, the ad server will potentially select different banner images for subsequent advertisement requests (steps 5–7).

Once a user shows interest in the displayed banner and clicks on it, the surrounding hyperlink will point back to the ad server where a script notes the click and sends a redirection message with the correct advertiser's site back to the originating browser. With a slight delay, the browser will request the correct product information page from the advertiser's server.

2.2. Current approaches

We can categorize current approaches to online advertisement into four categories [4]:

Untargeted.
Early systems and many small scale operations in use today simply target the broad Web audience in general. Ads are either fixed on a Web page for a certain time period and then manually updated, or a simple, random form of banner rotation is used [15].

Editorial.
Ad banners are targeted to a certain site or page topic. For example, advertisers on the Yahoo! [23] Web site can target their advertisement to any of the more than 100 thousand categories featured in the Web directory.

Targeted (filtered).
The most popular form of professional Web ad-

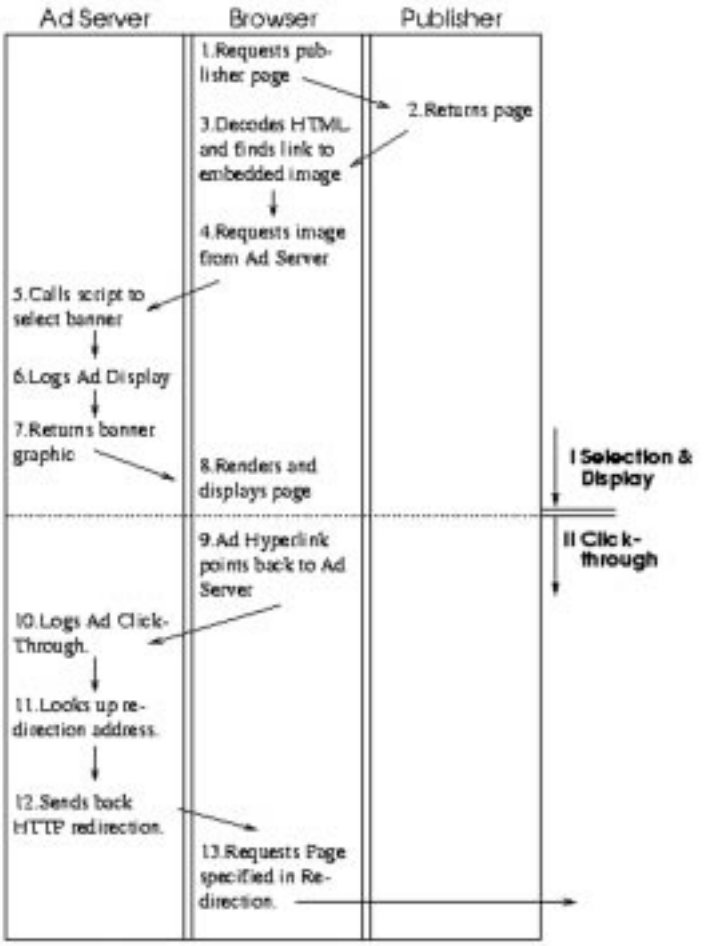

Fig. 2. Online advertisement: control and data flow. Embedded HTML images (step 3) and HTTP redirection (step 12) make it possible to separate page content and advertisement. After obtaining the page content from the publisher's Web server, the user's browser will request a dynamically selected advertisement from the ad server (part I). Once the user clicks on the advertisement (part II), the ad server will redirect the user to the appropriate site.

vertising today. Advertisers can specify targeting parameters such as the user's operating system or browser software, time constraints, country or even the Internet service provider (ISP) used. The selection mechanism on the ad server (step 5 in Fig. 2) analyzes the request and selects only those advertisements for placement that match the current situation.

Personalized.

Next generation advertisement systems use neural networks [2] and other proprietary learning methods [7] to allow personalized advertisement selection based on the browsing and interaction history of a particular user, as well as other demographic information.

2.2.1. Discussion

Using untargeted advertisement has two immediate advantages: it is simple to set up and it does not share any of the privacy concerns of personalized online advertisement. However, the drawback is often a much lower click-through rate since most of the displayed ads will be fairly unrelated to user interests. The traditionally rather homogeneous group of single, young males dominating the early Web has long since given way to a very diversified user

base, including almost 25% of users that are 50 years and older, as well as nearly 40% women [22, 9th survey], which makes it difficult to display a single advertisement that appeals to all its viewers.

Using editorial placement mechanisms, such as site specific advertising (for example on a site promoting the use of a certain operating system) or advertisement on topic-based pages (such as a Web directory), user interests directly overlap with the topic of the ad. While no user data needs to be collected, this form of advertisement placement still needs a high grade of monitoring in order to decide the best match between a certain advertisement and the page's content.

Targeted advertisement allows advertisers a greater level of control over who sees their advertisements and raises only minor privacy concerns, since usually no personal identifiable data is collected. However, systems like DoubleClick's DART [7] use unique identifiers to group the collected data by user [16]. In either case, advertisers have to constantly monitor and manually revise their targeting parameters in order to maximize the effectiveness of advertisement placement.

Finally, systems for personalized ad selection put the computer in charge of the constant adaptation of targeting constraints, but require a unprecedented level of user monitoring. Recent reports [3,10] already outline plans to connect such 'click-trail' databases with those of traditional mass-mailing companies, containing detailed demographics (income, age, gender) and buying-habit data about each user. Unsurprisingly, over 62% of Internet users do not trust sites collecting their online data [22, 9th survey] and more than two thirds ask for new laws on privacy [22, 8th survey].

2.3. Adaptive targeting

In our approach, we tried to avoid the common stereotype 'the more, the better' in favor of a less intrusive alternative. Our goal is to use two separate scales for advertisement customization: targeting *short-term* and *long-term* interests. Long-term interests are the goal of many of the personalization systems discussed above and become relevant when the user is simply browsing the Web and is not directly looking for a particular piece of information. Short-term interests on the other hand are dominant when the user is conducting a focused search for information by querying search services and Web Directories.

2.3.1. Observing short-term interests

One of the main sources for information about user short-term interests are the keywords the user submits to a search service. Alternatively, the URL of the page the user is currently viewing can act as a similar source of information for Web directories.

Both keywords and page URLs reflect user interest in a certain topic and can be used to customize the displayed advertisement. This information is said to be 'short-term' based, since users might have various reasons for looking up a certain article or Web page, that might or might not coincide with their regular interests.

Our 'short-term interests' approach, called 'adaptive targeting' is situated between the targeting-by-filtering and personalization approaches described above (see Fig. 3). Instead of amassing large amounts of information about the user as it is necessary for traditional approaches, our system can provide highly relevant advertisement with only a single piece of information (keywords or page URLs) by being able to automatically adapt to changes in correlation between this data and the actual advertisements.

2.3.2. Observing long-term interests

Soliciting detailed information about user interests does not need to be done behind a user's back. Many users are perfectly willing to share their preferences regarding a wide range of topics with Web businesses in exchange for value added services. The recent trend of personalized services on the Web such as customized newspapers [12,21] or personalized book, music [1] and movie [22] recommendations is a good indicator that many users are volunteering a substantial amount of information.

Instead of creating more and more sophisticated monitoring applications for Web advertising, we propose to use the readily available information stored in the user profiles of such services. In order to personalize advertisement to fit long-term user in-

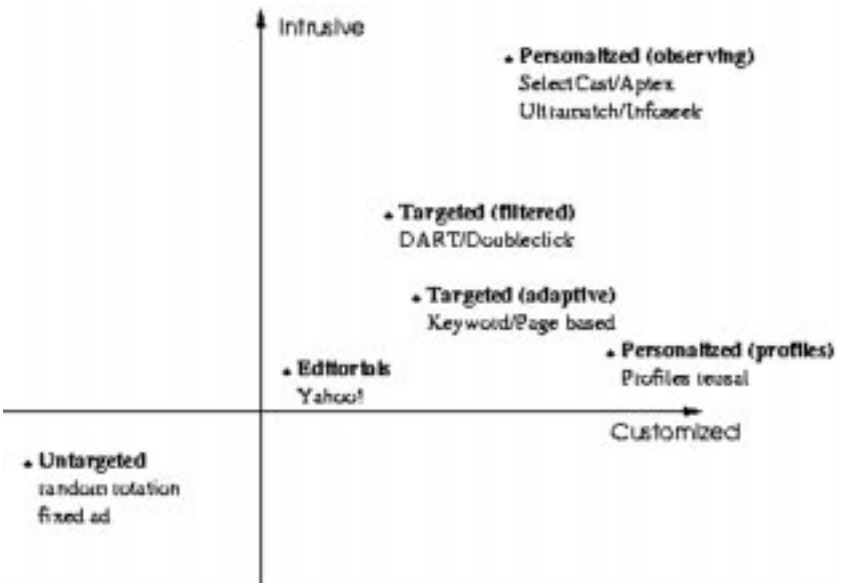

Fig. 3. Comparison of online advertisement systems. Traditional advertisement systems have to balance the tradeoff between improved customization and increased user surveillance. Separating short-term and long-term interests allows us to achieve higher customization with less intrusion of user privacy.

terests, such a system could completely rely on the personalization application to collect the necessary user profile information.

The current implementation of our adaptive targeting solution *ADWIZ* does not yet support long-term interest targeting, but several of our planned extensions to this extent are listed in Section 4.2. The following section will describe our implementation of a short-term interest based advertisement system, *ADWIZ*, in more detail.

3. The ADWIZ system

The *ADWIZ advertisement system* tries to capture immediate short-term user interests and select a suitable advertisement from the pool of available banners. Since only the keywords supplied to a search-service query are used to select the advertisement, the system does not need to use cookies to identify a particular user, nor does it store any user-related information in its database.

We will first give a very high level overview of the system's architectural components and its interfaces, and then proceed to describe its core components in more detail. Using empirical results obtained in our initial experiments we will then assess the system's effectiveness.

3.1. Architectural overview

The *ADWIZ* system consists of four principal components, as shown in Fig. 4: the *ad server (1)* handles the selection and actual delivery of the advertisement banner to the user; a separate *database server (2)* provides a central storage facility for all parts of the system, effectively decoupling each component and providing asynchronous communication; a *learning system (3)* runs periodically over aggregated performance statistics and dynamically calculates a set of display probabilities used by the advertisement selection system of the ad server; and an *administration server (4)* for inspecting and manipulating the database content such as advertisements and their properties, advertisement campaigns, and calculated display probabilities. In addition, our experimental setup contains a demonstration Web site which acts as a *publisher*'s search service and Web directory, and uses the functionality of our advertisement system for its customized ad display on both its pages and search results.

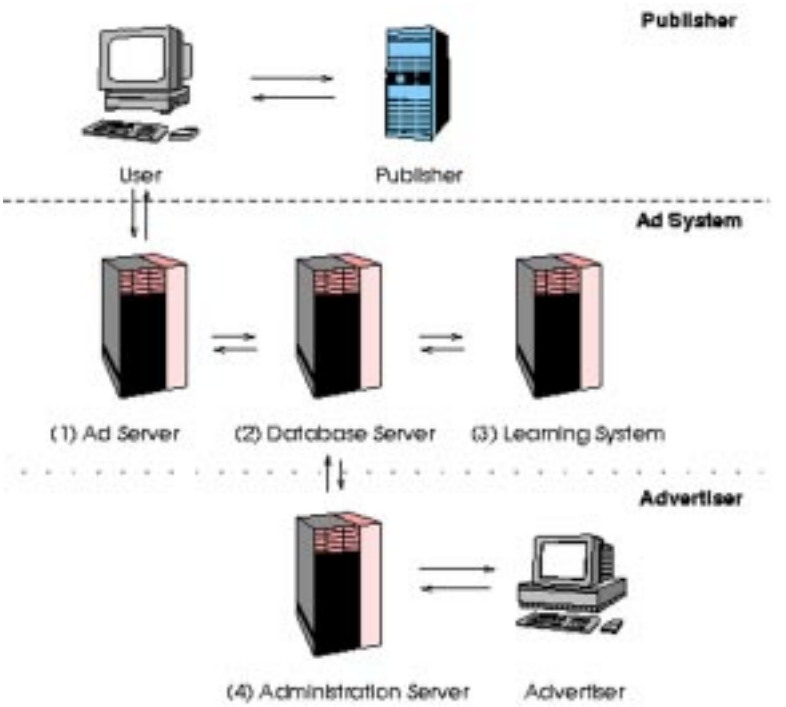

Fig. 4. System overview. The *ADWIZ* system consists of four parts: The *ad server* (1) handles requests for advertisements and selects a suitable banner ad for a search term or Web page; a *database server* (2) acts as a central storage facility and communication channel; the *learning system* (3) continuously updates the display probabilities for each advertisement stored in the database; while an *administration server* (4) allows advertisers (and publishers) to maintain and inspect the system status.

3.2. Interfaces

Fig. 4 also shows the two interfaces to the system (shown as computer screens in the picture). The front end interface allows users of the publisher's service to receive customized advertisements. An HTML construct on the publisher's Web page indicates that a graphical element (i.e. the ad banner) should be fetched from the advertisement system. The user's browser software will directly contact the linked server and request the graphics from the ad server (1), instead of from the publisher's Web server. The image will then be embedded into the publisher's page. From a user's point of view the interaction with the ad server is completely transparent.

Through its back end interface, advertisers (and/or publishers) can modify advertisements and their properties, schedule ad campaigns and obtain detailed performance reports. Advertisers can specify constraints similar to those offered in traditional advertisement systems, restricting display of an advertisement to certain target groups. Additionally, a keyword or particular Web page (or a percentage thereof) can be rented for an advertisement, resulting in a minimum guaranteed display rate for queries featuring this keyword or requests for that page.

3.3. Components

Since the administration server simply provides a convenient interface for inspecting and manipulating data stored on the database server, we will focus in the following sections on the three core components of the *ADWIZ* system: the ad server; the database server; and, in more detail, the learning system.

3.3.1. Ad server

The ad server is responsible for handling three basic modes of interaction with a user (as described in Fig. 2): requests for advertisement selection, re-

quests for advertisement image data and redirection requests resulting from user click-through. Its interface to a user's browser is realized as a set of CGI scripts that are invoked by a standard Web server software once a user requests an advertisement.

Since banner graphic display as well as user click-through handling are simple database lookups, we will focus our description here on the the ad server's principal module, the *selection engine*. The selection engine is responsible for selecting an advertisement from the pool of available banners for a given request (i.e. a search query or Web page request).

Fig. 5 shows the data flow within the selection engine that corresponds to step 5 in Fig. 2, the script call for selecting a banner. For simplicity we will assume a request for embedding an advertisement on the result page of a search service (i.e. a keyword based selection), although the same description would apply to the selection of a page based advertisement.

An *input decoding* module first decodes the parameters supplied through the CGI and extracts the set of keywords $f_1, f_2, \ldots, f_n$ that were used in the query to the search service. The *relevancy computation* module then uses a set of *weights* — display probabilities for each advertisement given a certain keyword — to compute a display probability $P(f_i)$ for each advertisement in the system, given the search terms $f_1, \ldots, f_n$.

The weights are periodically updated by the *learning system* using *performance data* collected by the system and a set of *display constraints* specified by the advertiser through the system's *advertisement management system*. Using this distribution, the *advertisement selection* module then chooses a particular advertisement ad_i and returns the graphical banner information back to the user's browser.

Note that the picture in Fig. 5 is simplified, since transactions between the management system (i.e. the administration server), the learning system and the ad server are all made through the database server.

In the click-through case (not shown) the ad server simply has to make a note in its performance table (stored in the database) that a selected ad has been 'successful' and then redirect the user's browser to the corresponding Web site specified for this ad. This redirection is again transparent to the user, who might in the worst case notice only a slight delay.

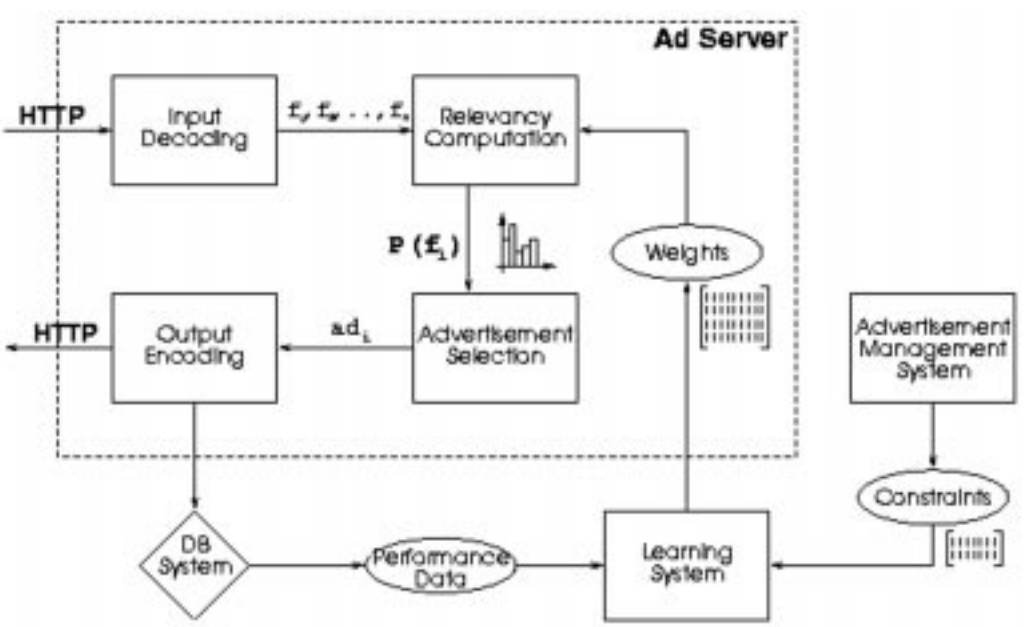

Fig. 5. The ad server's advertisement selection process. After extracting the search keywords, the ad server retrieves the corresponding weights and computes a display probability $P(f_i)$ for each advertisement in the system, given the search terms $f_1, \ldots, f_n$.

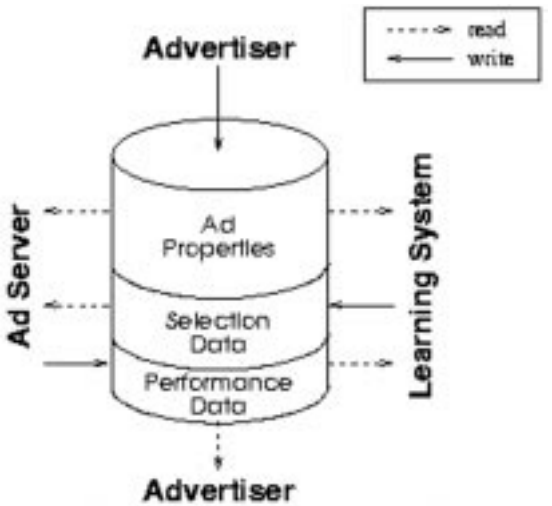

Fig. 6. Communication via the database server. After the advertiser sets all relevant *ad properties*, the learning system obtains campaign information and *performance data* and computes new *selection data* (i.e. weights). This information is used by the ad server during advertisement selection. Log information from the ad server is written back into the *performance data* tables, where it can be accessed by both the learning system and the advertiser.

3.3.2. Database server

The database server is the main communication facility for all parts of the advertisement system. Each component reads and writes data in the database tables, thus propagating changes in the system asynchronously to other modules. Fig. 6 shows the three types of tables — *ad properties* tables, *selection data* tables and *performance data* tables — and how they are used to connect the three core components of the advertisement system (compare with Fig. 4).

The advertiser sets a number of advertisement and campaign *properties* (i.e. when should which ad be displayed). The learning system continuously reads out the currently active advertisements from the *ad properties* and obtains *performance data* regarding each advertisement's click-through performance together with the corresponding keyword and Web page distribution (i.e. the number of times each keyword has been entered as a search term, and the number of times a particular Web page has been requested).

After computing the new display probabilities the learning system then updates the *selection data* tables, which are consulted by the ad server whenever a new advertisement needs to be selected. Given the id of the ad that should be displayed, the ad server can then get the banner image and necessary link information from the *ad properties* tables.

After each advertisement display or click-through redirection, the ad server will record the displayed or clicked advertisement and keyword pairs (or advertisement and page pairs) in the *performance data* tables, which the advertiser can use to monitor the system's performance.

3.3.3. The learning system

The learning engine system allows for both page-driven ads and (search) keyword-driven ads. In the following description of our learning engine, we again assume the latter setting for simplicity.

The basic idea is that the learning engine gathers statistics on the click-through rates for each advertisement–keyword pair, and then adjusts its ad display schedule in such a way that maximizes the total number of clicks.

One complication arises with this approach, however, due to the fact that, in actual advertisement contracts, a minimum number of displays is usually promised for each advertisement. Thus, simply displaying the 'best ads' for each keyword will not work, since poorly performing ads may never get displayed for any keywords. The click maximization problem we are dealing with is therefore an optimization problem with *constraints*. In particular, we can formulate the click maximization problem as a special form of linear programming problem as stated below.

Suppose that we are to probabilistically display m advertisements $A_1, \ldots, A_m$ depending on which of the search keywords $W_1, \ldots, W_n$ is input. We first need to calculate the desired display rate h_j for each ad A_j in the next period (until the learning engine is invoked next). This number can easily be computed by taking the promised number of displays for each advertisement and subtracting from it the number of times the ad has already been displayed. This figure is then divided by the number of remaining days in the advertisement's display period and normalized across all advertisements so that $\sum_{j=1}^{m} h_j = 1$.

We then estimate the input probability k_i of each keyword W_i and the click-through rate (clicks per displays) c_{ij} of each ad A_j on each keyword W_i

based on the past statistics. To calculate the expected keyword input probabilities we simply count the number of times each keyword has appeared in a query in the past (using a logarithmic decay factor) and normalize again so that $\sum_{i=1}^{n} k_i = 1$. During system operation each display and click-through of an advertisement is logged together with the corresponding search keywords that were used to select it. Thus dividing the number of clicks by the number of displays for each advertisement-keyword combination (i.e. A_j and W_i) gives us c_{ij}.

With these estimated quantities we set out to compute a display probability ('weight') d_{ij} that for a given keyword W_j tells our selection engine the probability with which a particular advertisement A_i from the list of available ads should be selected. The corresponding optimization problem we wish to solve can thus be formulated as the problem of setting the ad display rates d_{ij} for advertisement A_i on keyword W_j, so as to maximize the expected total click-through rate $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} k_i d_{ij}$ under the following constraints:

$$\sum_{i=1}^{n} k_i d_{ij} = h_j \quad (j = 1, \ldots, m), \tag{1}$$

$$\sum_{j=1}^{m} d_{ij} = 1 \quad (i = 1, \ldots, n), \tag{2}$$

$$d_{ij} \geq 0 \quad (i = 1, \ldots, n,\ j = 1, \ldots, m). \tag{3}$$

The first constraint ensures that the desired display rate h_j for each ad is satisfied, and the rest just ensures that the display probabilities are in fact *probabilities*, that is non-negative and sum to unity for each keyword.

Incidentally, many contracts come with so called 'keyword rental' and 'inhibitory keywords.' These can be naturally incorporated in the above formulation. For example, if ad A_j is 30% rented for keyword W_i, then we just need to replace the constraint (Eq. 3) for d_{ij} by the following:

$$d_{ij} \geq 0.3.$$

Inhibitory keywords could be handled similarly by replacing constraint (Eq. 3) for d_{ij} with $d_{ij} = 0$. But doing it this way has one problem: with too many of these inhibition constraints the optimization problem may become unsolvable and checking whether this is the case can be cumbersome. So we took an alternative approach of setting c_{ij} (the clicks-per-displays rate of advertisement A_j on keyword W_i that we computed from past statistics) to -1 when the advertiser wants to discourage the display of ad A_j for keyword W_i.

Notice how with this approach the total click rate is maximized when $d_{ij} = 0$, but that the system is still able to choose a non-zero value should display constraints make it necessary. In contrast, it is easy to check the feasibility of *rental* (i.e. non-zero) keyword constraints, since the problem is solvable if and only if $\sum_j d_{ij} \leq 1$ for all i and $\sum_i k_i d_{ij} \leq h_j$ for all j.

The above formulation of linear programming in fact belongs to the so called 'Hitchcock-type transportation problem' [6] (one can see this by substituting x_{ij} for $k_i d_{ij}$.) An efficient variant of the Simplex method is known for this class of problems, which uses (for m advertisements and n keywords) only $O(mn)$ space instead of $O(mn(m + n - 1))$ space required by the general Simplex method (see, for example, [13]). We adopted this method in our learning engine.

There is a subtle issue that need be addressed when we use the linear programming approach. This is the fact that an optimal solution of the above linear programming problem will tend to set most display probabilities d_{ij} to 0, and ad-keyword pairs that perform poorly at the beginning may never be displayed again. This way the confidence of estimation of the click-through rate for that ad-keyword pair will never improve. As one can imagine, it can happen that some ads that can potentially perform well on a keyword get unlucky and do badly at the beginning.

We address this issue by setting a minimum display rate for each ad-keyword pair, which is gradually lowered as a function of the sample size. More specifically, we replace the non-negativity constraint (Eq. 3) with the following:

$$d_{ij} \geq \frac{1}{2m\sqrt{D_{ij} + 1}} \tag{4}$$

where D_{ij} is the number of times A_j was displayed for W_i so far. Here the bound $1/(2m\sqrt{D_{ij} + 1})$ is derived from the standard deviation of the estimation of the click-through rate c_{ij}.

3.4. Empirical evaluation

We evaluated our learning engine using artificially generated data. We constructed probabilistic models of users, namely a probability distribution $D(i)$ for the keyword generation and a conditional probability model $C(i, j)$ for click-through rates conditioned by a keyword and an ad, and used a random number generator to simulate them. We ran our learning engine in the artificial environment thus constructed and measured its performance.

More specifically, the following trial was repeated one million times, while updating the display probabilities d_{ij} every 3125 trials by feeding the statistics obtained up to that point to the learning engine:

(1) Generate a keyword W_i randomly according to probability $D(i)$.
(2) Select an ad A_j randomly according to the display probabilities $\{d_{i1}, \ldots, d_{im}\}$.
(3) Decide whether A_j is clicked randomly according to probability $C(i, j)$.

The display rates d_{ij} were set to be uniform initially and changed over time as the learning engine adjusted them to optimize the total click-through rate. The constant click-through probabilities $C(i, j)$ were manually set using a semi-automated process described in the next paragraph.

In the experiment that we report here, we used models that consisted of 32 ads and 128 keywords. The 128 keywords were divided into 32 groups of four keywords, with each group appearing with equal probability. Within each group, four keywords appear with different probabilities (with ratios 1:2:3:4). First, a click-through rate assignment to the 32 ads was created by hand, including high, moderately high and low click rates. Then for each of the 32 keyword groups, we created a different pattern of assignment by rotating the original assignment pattern and adding some noise. Within each keyword group, the basic assignment pattern was further modified

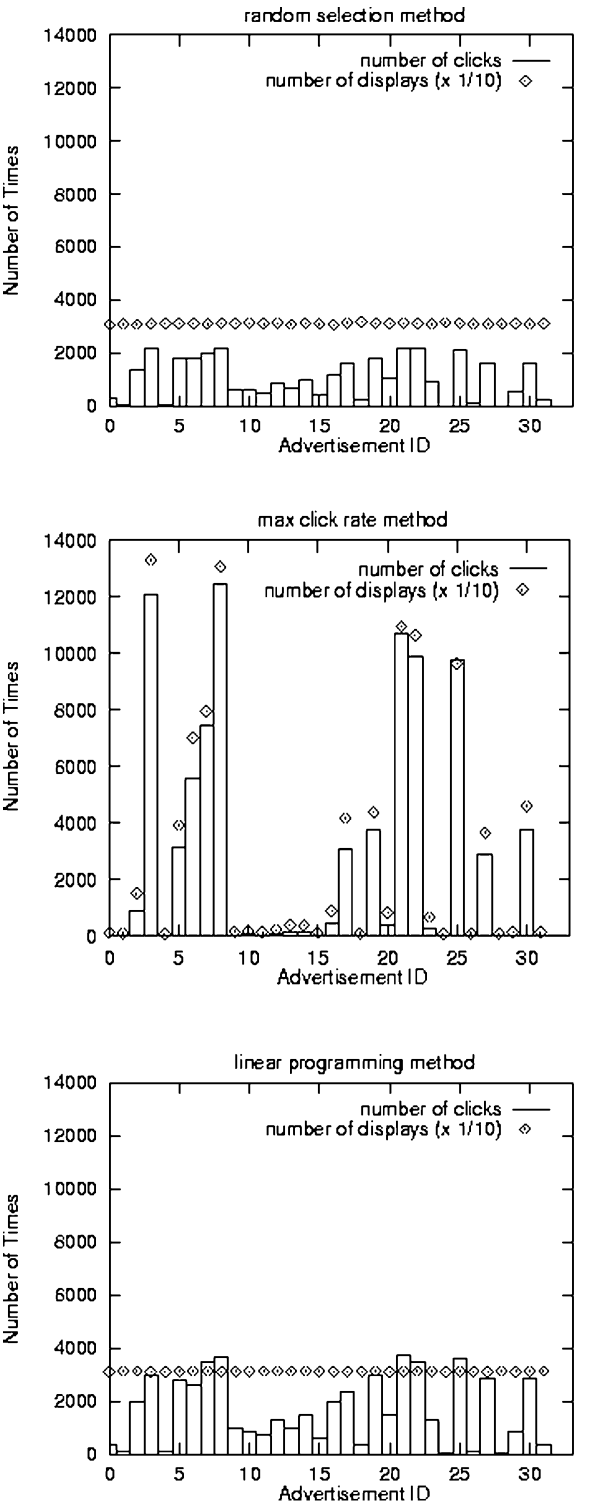

Fig. 7. The total number of displays and clicks per ad. The diamond shaped dots indicate (one tenth of) the number of displays, and the bars indicate the number of clicks. Although the max click rate achieves the highest total click-through rates, close examination will reveal that with this method half the ads did not get displayed much.

by multiplying the probabilities by a random real number in $a_j \in [0, 1]$ for each ad A_j. We had no constraints involving rental or inhibitory keywords.

In our evaluation, we compared our learning method with two simple methods: the method of selecting an ad randomly (the random selection method) and the method of selecting an ad with the maximum click-through rate for the given keyword in the current data (the max click rate method).

Fig. 7 shows the results of these experiments. The results were obtained by averaging over five runs in each case. The diamond shaped dots indicate one tenth of the number of displays, and the bars indicate the number of clicks. The first figure is the result using the random selection method, the second figure is for the max click rate method, and the last figure is for our method.

While it is true that the max click rate method achieves the highest total click-through rates, close examination will reveal that with this method half the ads did not get displayed much. Thus, this method is most likely not usable in practice. The number of displays for the ads is balanced for both the random selection method and our method, but the total number of clicks yielded by our method is significantly higher. Fig. 8 plots the cumulative total click-through rates achieved by each the three methods over time.

We also ran experiments to evaluate the effect of the modification we described earlier of assuring a minimum number of displays for each ad-keyword pair Eq. 4. Fig. 9 plots the click-through rates achieved by the version with the modification and the version without it. The click-through rate using the modified version rises more slowly at the beginning since it is doing more exploration, but in the end, it surpasses the vanilla version by a significant margin.

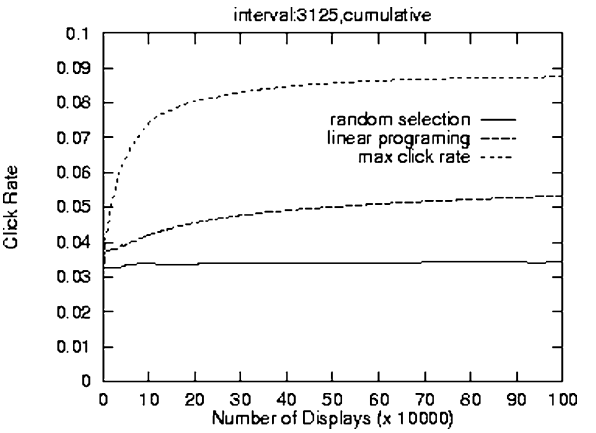

Fig. 8. Plots of the cumulative total click-through rates of three methods. The max click rate method achieves the highest click-through rate but at the price of neglecting poorly performing ads. Using our linear programming approach we can increase the total click-through rate while keeping overall display rates balanced (compare with Fig. 7).

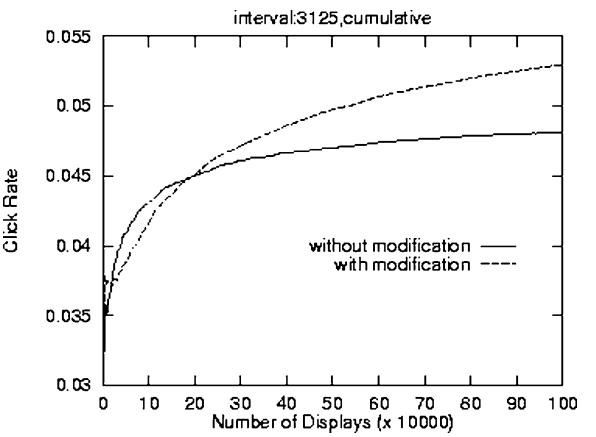

Fig. 9. Plots of the total cumulative click-through rates of two versions of our method. When using a modification that prevents our optimization method from setting too many weights to zero we are able to further increase its total click-through rate.

4. Conclusions

In order to raise click-through rates for Web advertisement we do not have to abandon a user's privacy. Our experiments indicate that a simple learning technique based on keywords and page URLs already has the potential of significantly increasing the relevance of advertisement banners.

By using an automated, constraint based learning method we are able to:

- allow publishers of Web services to increase their inventory space value without having to attend to the actual targeting and placement details;
- give advertisers full control over their advertisement targeting while maximizing the effectiveness of each banner over all unconstrained features; and

- lessen the annoyance of online advertisement for users by providing customized, context sensitive ad banners without invading consumer privacy.

Although a simple selection method that always uses the best performing advertisement achieves the highest total click-through rate in our experiments, the minimum display constraints found in many advertisement contracts today render such strategies useless in practice. Our first prototype has encouraged us to continue our work on unintrusive advertisement customization methods.

4.1. Related work

With the growing importance of advertisement for the Web there has been an increasing amount of research in this area.

[14] describes a system which separates advertisements and publishers' Web sites by introducing an advertisement agent. The agent sits between advertisers and the user's browser and merges banner advertisement directly into the currently viewed page, independent of the page itself. A number of companies have introduced similar systems, for example displaying advertisement while the user waits for Web content to download, but these have so far failed to gain widespread acceptance.

A system developed by [4] uses explicit interest solicitation, both through a special form-based interface and by using negative feedback controls next to each displayed banner advertisement. Although this approach is both privacy friendly and promises a high potential for personalization, a real-world deployment has yet to show how much effort the user is willing to make in order to receive customized advertisements.

Other studies [5] suggest that the value of banner advertising goes beyond direct click-through responses, influencing the user even when he or she does not click on the surrounding hyperlink. However, so far no advertiser has adopted any other method for measuring the effectiveness of an ad, rendering click-through maximization performed by our system still the only available benchmark. Finally, [19] reports that other forms of advertisement, such as site sponsorship, might be better suited to get user attention than simple banner ads. Although the level of such promotional advertisement continues to increase, the Internet Advertisement Bureau still sees banner advertisement as the dominant form of online advertising [24].

4.2. Future work

We stand at the very beginning of our project, and many issues are currently unresolved.

Scaling Up.
In order to be usable in a real-world setting, our keyword based system has to be able to handle hundreds of thousands of keywords, including proper names, foreign words and misspellings. Proper clustering techniques lie at the heart of such scaling, and we will need to investigate ways to reduce the complexity of the search space the learning system has to cover, while preserving the quality of the available data.

Real-world deployment.
In order to validate our hypotheses — effective advertisement targeting is possible without invading user privacy — we will need to move our advertisement system from a laboratory environment to a real-world server. Many of the problems associated with such a step are rather technical (guarantee server uptime, provide efficient user interfaces for both advertisers and publishers, etc.) and seem to distract us from the research itself. However, building systems that can perform in the real world presents the ultimate validation of any technology designed to improve existing shortcomings.

Long-term interests.
The methods described in this paper are targeted to detect *short-term* user interests while interacting with a search service or Web directory. In a second step, we plan to combine this system with a more user-centric personalization system that is able to detect long standing user interests and hobbies. Using both short- and long-term interests simultaneously, our goal is not to monitor a user's complete online habits, but to reuse an *existing* profile of a personalized system that has been customized with full user consent and cooperation. This reflects the approach taken in [4], with the important distinction that the user does not personalize the advertisements itself, but rather interacts with the personalized system in a

standard fashion while simultaneously improving the relevance of the displayed advertisements.

References

[1] Amazon.com, Recommendation Center, http://www.amazon.com

[2] Aptex Software Inc., SelectCast Affinity Server, http://www.aptex.com/products-selectcast-ads.htm

[3] Associated Press, Personal habits gathered for use on the Internet, CNN interactive, Atlanta, GA, August 16, 1998, available at http://www.cnn.com/TECH/computing/9808/16/website.privacy/

[4] P. Baudisch and D. Leopold, User-configurable advertising profile applied to Web page banners, in: Proc. 1st Berlin Economics Workshop, Berlin, Germany, October 1997, see also the Postscript file called 'baudisch1997bUserConfigurableAdvertisingProfiles.ps' at http://www.darmstadt.gmd.de/~baudisch/Publications/

[5] R. Briggs and N. Hollis, Advertising on the Web: Is there response before click-through? Journal of Advertising Research March/April (1997) 33–45.

[6] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, 1963.

[7] DoubleClick Inc., DART Advertisement System, home page at http://www.doubleclick.com.

[8] R. Fielding et al., The Hypertext Transfer Protocol — HTTP/1.1 Internet Draft, August 1998, available at http://www.w3.org/Protocols/HTTP/1.1/draft-ietf-http-v11-spec-rev-04.txt

[9] S. Hansell, Big Web sites to track steps of their users, The New York Times, August 16, 1998, available at http://www.nytimes.com/library/tech/98/08/biztech/articles/16data.html

[10] P. Joseph, On-line advertising goes one-on-one, Scientific American, December 1997, available at http://www.sciam.com/1297issue/1297cyber.html

[11] Jupiter Research report cited by Cyberatlas, February 1997, See the Cyberatlas Market Forecast report at http://www.cyberatlas.com/segments/advertising/market_forecast.html

[12] Kobayashi, Introduction to Linear Programming Methods, Sangyo Tosyo, 1980 (in Japanese).

[13] Y. Kohda and S. Endo, Ubiquitous advertising on the WWW: merging advertisement on the browser, in: 5th Int. World-Wide Web Conference, Paris, France, May 1996, see also http://www5conf.inria.fr/fich_html/papers/P52/Overview.html

[14] K. Lang, NewsWeeder: learning to filter netnews, in: Proc. 12th Int. Machine Learning Conference (ICML '95), Lake Tahoe, CA, Morgan Kaufmann, San Francisco, pp. 331–339.

[15] LinkExchange Network, http://www.linkexchange.com

[16] Z. Moukheiber. DoubleClick is watching you, Forbes Magazine, April, 1996, available at http://www.forbes.com/forbes/110496/5811342a.htm

[17] Network Wizards, Survey results July 1998, Internet Domain Survey, July 1998, available at http://www.nw.com/zone/WWW/top.html

[18] D. Raggett et al. (Eds.), The HTML 4.0 Specification, W3C Recommendation, April 1998, available at http://www.w3.org/TR/REC-html40/

[19] K. Risden et al., Interactive advertising: patterns of use and effectiveness, in: Proc. CHI '98, Los Angeles CA, April 1998, pp. 219–224.

[20] J. Rogers (director), GVU's WWW User Survey, Graphic, Visualization, and Usability Center, Georgia Tech, April October 1994–1998, available at http://www.gvu.gatech.edu/user_surveys/

[21] H. Sakagami, T. Kamba and Y. Koseki, Learning personal preferences on online newspaper articles for user behaviors, in: Proc. 6th Int. World Wide Web Conference, 1997, pp. 291–300.

[22] B. Sarwar, J. Konstan, A. Borchers, J. Herlocker, B. Miller and J. Riedl, Using filtering agents to improve prediction quality in the GroupLens Research Collaborative Filtering System, in: Proc. 1998 Conf. on Computer Supported Cooperative Work, November 1998, http://www.movielens.umn.edu/

[23] Yahoo! Inc., Yahoo! Web Directory, http://www.yahoo.com

[24] ZDnet Interactive report, Banner ads alive and still tops, cited by NUA Internet Surveys, June 18, 1998, available at http://www.nua.ie/surveys/index.cgi?service=view_survey and surve_number=821 and rel=no.

[25] ZDnet Interactive report, AOL hangs up telemarketing plan, NEWSwatch, August 1997, available at http://www.zdnet.com.au/zdimag/news/199707/28/news6.html

Marc Langheinrich received a masters degree in computer science from the University of Bielefeld, Germany, in 1997. Starting in the fall of 1995, he spent a year as a Fulbright Scholar at the University of Washington, where he also completed his thesis work in the fields of information retrieval and software agents. In the fall of 1997 he joined NEC Research in Japan and has since been working on projects involving personalization and electronic commerce.

Atsuyoshi Nakamura received his B.S. and M.S. degrees in computer science from the Tokyo Institute of Technology in 1986 and 1988. He is presently an assistant manager in the C&C Media Research Laboratories of NEC Corporation. His interests have been in the area of machine learning, especially computational learning theory and information filtering.

Naoki Abe received the B.S. and M.S. degrees from the Massachusetts Institute of Technology in 1984, and the Ph.D. degree from the University of Pennsylvania in 1989, all in Computer Science. After holding a post doctoral researcher position at the University of California, Santa Cruz, he joined NEC Corporation in 1990, where he is currently principal researcher in the C&C Media Research Laboratories. He is also visiting associate professor in the department of computational intelligence and systems science at the Tokyo Institute of Technology. His research interests include theory of machine learning and its applications to various domains, including internet information mining and navigation.

Tomonari Kamba received his B.E. and M.E., and Ph.D. degrees in Electronics from the University of Tokyo in 1984, 1986 and 1997 respectively. He joined NEC Corporation in 1986, and he has been engaged in user interface design methodology, multimedia user interface, software agent, and Internet information service technology. He was a visiting scientist at the Graphics, Visualization and Usability Center at the College of Computing, Georgia Institute of Technology from 1994 to 1995. He is now a principal researcher in the C&C Media Research Laboratories, NEC Corporation. Dr. Kamba is a member of ACM SIGCHI and the Information Processing Society of Japan.

Yoshiyuki Koseki received his B.S. degree in computer science from the Tokyo Institute of Technology in 1979 and his M.S. degree in computer science from the University of California at Los Angeles in 1981. He received his Ph.D. in System Science from Tokyo Institute of Technology in 1993. He joined NEC Corporation in 1981, and has since been engaged in research on artificial intelligence, expert systems, visual programming, information visualization, and World-Wide Web-based agent technology. He is now a Senior Research Manager in NEC's C&C Media Research Laboratories and is a member of the IEEE, the Japanese Society for Artificial Intelligence, and the Information Processing Society of Japan.