# SECTION I

# OVERVIEW OF E-COMMERCE RESEARCH CHALLENGES

# 1

# STATISTICAL CHALLENGES IN INTERNET ADVERTISING

DEEPAK AGARWAL
*Yahoo! Research, Santa Clara, CA, USA*

## 1.1 INTRODUCTION

Internet advertising is a multi-billion-dollar industry, as is evident from the phenomenal success of companies like Google, Yahoo, Microsoft, and continues to grow at a rapid rate. With broadband access becoming ubiquitous, Internet traffic continues to grow in both volume and diversity, providing a rich supply of inventory to be monetized. Fortunately, the surge in supply has been accompanied by an increase in demand, with more dollars being diverted to Internet advertising relative to traditional advertising media like television, radio, and newspapers.

Marketplace designs that maximize revenue by exploiting billions of advertising opportunities through efficient allocation of available inventory are the key to success in this scenario. Due to the massive scale of the problem, an attractive way to accomplish this is by learning the statistical behavior of the environment through the huge amounts of data constantly flowing into the system. Furthermore, automated learning reduces overhead and has a low marginal cost per transaction, making Internet advertising a lucrative business. However, learning in these scenarios is highly nontrivial and gives rise to a series of challenging statistical problems, including prediction of rare events from massive amounts of high-dimensional data, experimental designs to learn emerging trends, and protecting advertisers by constantly monitoring traffic quality. In this chapter, I provide a perspective on some of the statistical challenges through illustrative examples.

*Statistical Methods in e-Commerce Research*. Edited by W. Jank and G. Shmueli

## 1.2 BACKGROUND

Web advertising supports a broad swath of today's Internet ecosystem, with an estimated $15.7 billion in revenues for 2005 (www.cnnmoney.com). Traffic and content on the Web continue to grow at a rapid rate, with users spending a larger fraction of their time on the Internet. This trend has caught the eye of the advertising industry, which has been diverting more advertising dollars to the Internet. Thus, revenue continues to grow, both in the United States and in international markets.

Currently, two main forms of advertising account for a large fraction of the total Internet revenue. The first, called Sponsored Search advertising, places ads on result pages from a Web search engine like Google, Yahoo!, or MSN, where the ads are driven by the originating query. In contrast to these search-related ads, the second, more recent advertising mechanism, called Contextual Advertising or Content Match, refers to the placement of commercial text ads within the content of a generic Web page. In both Sponsored Search and Content Match, usually there is a commercial intermediary, called an *ad network*, in charge of optimizing the ad selection, with the twin goals of increasing revenue (shared by the publisher and the ad network) and improving the user's experience. Typically, the ad network and the publisher are paid only when the user visiting the Web page or entering keywords in a query box *clicks* on an advertisement (often referred to as the *pay-per click* (*PPC*) model. For instance, both Google and Yahoo! have such ad networks in the context of Content Match which cater to both large Web publishers (e.g., AOL, CNN) and small Web publishers (e.g., owners of blog pages). Introduced by Google, Content Match provides an effective way to reward publishers who are creators of popular content. In Sponsored Search, most major search engines often play the twin roles of publisher and ad network; hence, they receive the entire proceeds obtained from clicks on advertisements.

Yet another form of Internet advertising that still has a lucrative market is the display of graphical or banner ads on content pages. For instance, this advertising model is used extensively by Yahoo! on its properties like Mail, Autos, Finance, and Shopping. One business model charges advertisers by the number of displays or *impressions* of advertisements instead of clicks. In general, this is a rapidly evolving area and there is scope for new revenue models.

Of the three forms of Internet advertising just discussed, Sponsored Search typically display ads that are more relevant since the keywords typed by the user in the query box are often better indicators of user intent. In Content Match, user intent is inferred indirectly from the context and content of the page being visited; hence, the ads being shown typically tend to be less relevant than those on Sponsored Search. For banner ads, intent information is typically weaker compared to both Sponsored Search and Content Match; it is generally used by advertisers as a brand awareness tool. In both Sponsored Search and Content Match, since advertisers are charged only when ads are clicked (the amount paid is often called *cost per click* or *CPC*), the clicks provide a meterable way to measure user feedback. Also, advertisers can monitor the effectiveness of their Sponsored Search or Content Match advertising campaigns by tracking *conversions* (sales, subscriptions, etc.)

that accrue from user visits routed to their Websites through clicks on ads in Sponsored Search and Content Match. For banner ads, advertisers are typically charged per display (also called *cost per milli* (thousand) or *CPM*). As expected, CPC for Sponsored Search is typically higher than that for Content Match, and the CPM model in banner ads typically yields lower revenue than Sponsored Search and Content Match per impression. Finally, all three advertising mechanisms are automated procedures with algorithms deciding what ads to display in which context. Automation enables the system to work at scale, with low marginal cost, and leads to a profitable business.

The rest of the chapter is organized as follows. We begin by providing a brief high-level overview of search engines in Section 1.3. In Sections 1.4 and 1.5, we provide a brief description of ad placement in the context of Sponsored Search and Content Match, followed by a detailed description of the important statistical problem of estimating click-through rates. Section 1.6 describes the problem of measuring the quality of clicks received in Sponsored Search and Content Match, also known as *click fraud* in popular media. In Section 1.7, we discuss next-generation search engines and the challenges that arise thereof. We conclude in Section 1.8.

## 1.3 SEARCH ENGINES

This section provides a brief high-level overview of how search engines work. This is useful in understanding some of the statistical challenges we discuss later in the chapter.

Before delving into the details of search engine technology, we provide a brief description of how the World Wide Web (WWW) works. In the most common scenario, a user requests a webpage by typing in an appropriate URL on the web browser. The page is fetched via an http (protocol to transmit data on the WWW) request issued by the user's web server (typically, a machine running a software program called Apache) to the destination web server. The transmission of data takes place through a complex mechanism whereby another server, called the *Domain Names Server* (DNS), translates the URL, which is in human-understandable language, into an IP address. The IP address is used to communicate to the destination web server through special-purpose computers called *routers*. With the availability of broadband technology, this entire mechanism is amazingly fast, typically taking only a few milliseconds. Once the destination server receives the request, it transmits the requested page back to the user's web server via the routers.[1] The files requested are mostly written in Hypertext Markup Language (HTML) (files in other formats, like ppt and pdf, can also be requested), in which tags are used to mark up the text. The tags enable the browser to display the text content on the requested HTML page. The HTML page contains a wealth of information about the webpage and is extremely useful in extracting features that can be used for various statistical modeling tasks. Among other things, it contains *hyperlinks* that are

[1]A complete description of how this transfer takes place is beyond the scope of this chapter.

typically URLs providing links to other pages. The hyperlinks are extremely useful and have been used for various modeling tasks, including computation of the popular PageRank algorithm (Page et al. 1998). Each hyperlink is annotated with text called *anchor text*. Anchor text provides a brief, concise description of pages and hence serves as a useful source from which important features can be extracted. For instance, if anchor text from several hyperlinks pointing to a page agree closely on the content, we get a fairly good idea of page content.

We now provide a brief overview of how search engines work. There are three main steps: (a) continuously getting updated information on the WWW by running automatic programs called *crawlers* or *spiders*; (b) organizing content in retreived pages efficiently, with the goal of quick retrieval during query time (called *indexing*); (c) at query time, retrieving relevant pages and displaying them in rank order, with the more relevant ones being ranked higher. This has to be done extremely fast (typically in less than a few milliseconds).

### 1.3.1 Crawler

The Web is huge and diverse, and storage space and network bandwidth are finite. Hence, it is not feasible for search engines to keep a current copy of the entire WWW. Thus, the crawler has to be smart in selecting the pages to crawl. Typically, the search engine starts with a seed of domain names, crawls their home pages, crawls the hyperlinks on these pages, and recurses. The problem is compounded since the arrival rate of new content on the Web is high, and it is important to keep up with fresh content that might be of interest to users. There are several other technical issues that make crawling difficult in practice. Servers are often down or slow, hyperlinks can put the crawler into cycles, requests per second on an individual website are limited due to politeness rules, some websites are extremely large and cannot be crawled in a small amount of time while obeying the politeness rules, and many pages have dynamic content (also referred as the *hidden web*) which can be only retrieved by running a query on the page. Prioritizing page crawls is a challenging sequential design problem. Note that sampling of pages here is more involved than traditional sequential design due to the graph structure induced by hyperlinks. The sequential design should be able to discover new content efficiently under all the constraints mentioned above to keep the index fresh. Also, we want to minimize the number of recrawls for pages that do not change much. In other words, we may want to recrawl pages based on their estimated change frequency (see Cho and Ntoulas 2002; Cho and Garcia-Molina 2003 for details). However, crawling high-frequency pages may not be an optimal strategy to discover new content. A naive strategy of crawling all new pages may also be suboptimal. This is so because old low-change-frequency pages may contain links to other old but high-change-frequency pages which, in turn, provide links to a large number of new pages. What is the best trade-off between recrawling old pages and crawling new pages? Detailed discussion of this and some other issues mentioned above can be found in Dasgupta et al. (2007), along with an initial formulation using the *multi–armed bandit* framework, perhaps one of the oldest formulations of sequential design popularized in

statistics by seminal works of Gittins (1979) and Lai and Robbins (1985). The main idea in multi–armed bandit problems is to devise an adaptive sampling procedure which will identify the best of $k$ given hypothesis using a small number of samples. The sampling procedure at any given time point is a rule which depends on the outcomes that have been observed so far. Adaptive sequential designs are routinely used in statistics (Rosenberger and Lachin 2002) in the context of clinical trials, but in this context the problem is high-dimensional, with constraints imposed by the structure of the hyperlink graph. Moreover, the objective function here is quite different from the ones used in clinical trials literature. This is a promising new area of research for statisticians with expertise in experimental design and sampling theory.

Once we crawl a page, the next question is, what information should we store about the page? The typical information we store includes words in the title, body, inlinks, outlinks, anchor text, etc. Some pages might be long, and storing every word might not add much value. Thus, the statistical problem here is to characterize a set of sufficient statistics that capture most of what the page is about. This is also referred to as *feature extraction* in machine learning and data mining. Of course, computing such sufficient statistics would require a statistical model, which may be driven by editorial judgments on a small set of pages and click feedback obtained on a continuous basis when pages are shown by the search engine in response to queries. Several such models based on ideas from machine learning, data mining, and statistics are currently used by search engines, the details of which are often closely guarded secrets. However, there is substantial scope for improvement. The abstract statistical problem in this context can be stated as follows: Given an extremely large number of features and two response variables, the first one being more informative but subjective and costly to obtain and the second one being less informative but inexpensive to obtain, how does one devise statistical procedures that can do effective variable selection? The problem gets even more complex since a nonignorable fraction of pages are affected by spam, which is perpetuated mainly to manipulate the ranking of pages by search engines.

### 1.3.2 Indexing

Once content from crawled pages is extracted, one needs to organize it to facilitate fast lookup at query time. This is done by creating an *inverted index*. In general, this is done by first forming a dictionary of features and, for each feature, associating all document identities that contain the feature. In reality, the index is huge and has to be spread across several machines, with clever optimization tricks used to make the lookup faster.

### 1.3.3 Information Retrieval

The last step consists of procuring documents from the index in response to a query and displaying them in a rank-ordered fashion. This is an active area of research in computer science, SIGIR being a major conference in the area. We refer the reader to Manning et al. (2008) for an introduction. At a high level, the search engine

looks up words contained in the query in the inverted index and retrieves all relevant pages. It then rank orders the documents and displays them to the user. The entire process has to be fast, typically taking a few milliseconds. The ranking is based on a number of criteria, including the number of words on the page that match the query, the location of matches on the page, frequency of terms, page rank (which provides a measure of the influence the page has on the hyperlink graph), the click rate on the page in the context of a query, editorial judgments, etc. Again, creating algorithms to combine information from such disparate sources to provide a single global ranking is a major statistical challenge that determines the quality of the search engine to a large extent. In general, algorithms are trade secrets and are not revealed. Also, making changes to algorithms is routine but evaluating the effect of these changes on quality is of paramount importance. One effective way to solve this problem is through classical experimental design techniques (e.g., factorial designs).

## 1.4 ESTIMATING CLICK-THROUGH RATES

We now provide a brief high-level description of procedures that are used to place ads both in Sponsored Search and Content Match. We then introduce an important problem of estimating click-through rates (CTR) in both Sponsored Search and Content Match and discuss some statistical challenges.

In Sponsored Search, placement of ads in response to a query depends on three factors: (a) relevance of the ad content to the query, (b) the amount of money an advertiser is willing to pay per click on the ad, and (c) the click feedback received for the ad. Relevance is determined by keywords that are associated with ads and decided by advertisers a priori when planning their advertising campaigns. Along with the keyword(s), the advertiser also places a *bid* on each ad, which is the maximum amount he or she is willing to pay if the ad is clicked once. Typically, advertisers also specify a *budget*, i.e., an upper bound on the amount of money they can spend. As with search results, candidate ads to be shown for each query are obtained by matching keywords on ads with the query. The exact forms of matching functions are trade secrets that are not revealed by ad networks. In general, there is an algorithm that determines if the keyword(s) match(es) the query exactly (after normalization procedures like removing stop words, stemming, etc.) and a series of algorithms which determine if there is a close conceptual match between query and keyword(s). The candidate ads are then ranked according to revenue ordering, that is, according to a product of the bid and a relevance factor related to the expected CTR of the ad. Thus, an ad can be highly ranked if it is highly relevant (i.e., CTR is high), and/or the advertiser is willing to pay a high price per click. The rankings determine the placement of ads on the search engine page: the top-ranked ad is placed in the top slot, and so on. The CTR, of ads placed higher on a page is typically higher than the CTR of ads placed in lower slots. The actual amount paid by an advertiser when a click occurs is determined by an extension of the second price auction (Varian 2007; Edelman et al. 2006) and in general depends on the CTR and bid of

the ad that is ranked directly below the given ad. In the most simple form, if all CTRs are equal, an advertiser's payment per click is the bid of the next highest bidder.

In Content Match, every showing of an ad on a webpage (called an *impression*) constitutes an event. Here, among other things, matching of ads is based on the content of the page, which is a less precise indicator of user intent than a query provided in Sponsored Search.

In both Sponsored Search and Content Match, estimating the CTR for a given (query/page, ad) pair in different contexts is a challenging statistical problem. The context may include the position on the page where the ad is placed, user geography derived from the ip address, other user features inferred from browsing behavior, time-of-day, day-of-week, day-of-year information, etc. A rich class of features is also available from the query/page and the ad. The estimation problem is challenging for several reasons, some of which are as follows:

- *Data sparsity*: The feature spaces are extremely large (billions of query/pages, millions of ads, with great diversity and heterogeneity in both query/pages and ads) and the data are extremely *sparse*, since we observe only a few interactions for a majority of query/page-ad feature pairs.
- *Rarity of clicks*: The CTR, defined as the number of clicks per impression (number of displays) for a majority of page-ad feature pairs, is small.
- *Massive scale*: The number of observations available to train models is huge (several billions), but one generally has access to a grid computing environment. This provides a statistical computing challenge of scaling up computations to fit sophisticated statistical models by harnessing the computing power available.
- *Ranking*: Although we have formulated the problem as estimating CTRs, in reality what is needed is a method that can rank the ads. Thus, transforming the problem to predict a monotone function of CTR to produce a rank-ordered list is a good approach and opens up new opportunities to obtain *clever* approximations.

To provide an idea of the sparsity inherent in the data, Figure 1.1a shows the frequency of (page, ad) pairs and Figure 1.1b shows the same distribution for a subset of impressions where a user clicks on the ad being shown on the page from a Content Match application. Clearly, an overwhelming majority of (page, ad) pairs are extremely rare, and a small fraction account for a large fraction of total impressions and clicks. Naive statistical estimators based on frequencies of event occurrences incur high statistical variance and fail to provide satisfactory predictions, especially for rare events. The usual procedure involves either removing or aggregating rare events to focus on the frequent ones. While this might help estimation at the "head" of the curve, the loss in information leads to poor performance at the "tail." In Internet advertising, the tail accounts for several billion dollars annually, making reliable CTR estimation for tail events an important problem.

Replacing pages and ads with their features and fitting a machine learning model is an attractive and perhaps the most natural approach here. In general, a machine

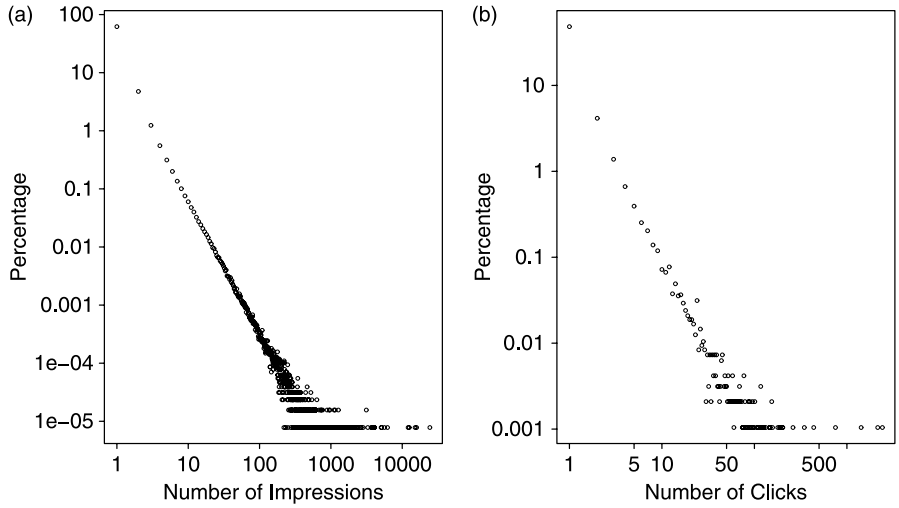


**Figure 1.1** (a) Distribution of impression events and (b) click events. Plots are on log-log scale but ticks are on the original scale; 99.7% of impression events had no clicks.

learning model with a reasonable number of features performs well at the head; the problem begins when one starts fitting features to "chase" the tail. A large fraction of features tend to be sparse, and we may end up overfitting the data. One solution to this problem is to train machine learning models on huge amounts of data (which are available in our context), but that opens up the problem of scaling computations. Typically, one has access to a grid computing environment which is generally a cluster of several thousand computers that are optimized to perform efficient distributed computing. However, algorithms for fitting machine learning and statistical models were not developed to perform distributed computing, and hence the subject needs more research.

The rarity of clicks with sparseness of features makes the problem even more challenging. There is substantial literature on machine learning for predicting imbalanced or rare response variables (Japcowicz 2000; Chawla et al. 2003, 2004). Most of the approaches rely on sampling the majority class to reduce the imbalance. In statistics, the paper by King and Zeng (2001) discusses logistic regression with rare response. The authors note that with extreme imbalance, the logistic regression coefficients can be sharply underestimated, and suggest sampling and bias correction as a remedy. Recently, an interesting paper by Owen (2007) derived the limiting behavior of logistic regression coefficients as the amount of imbalance tends to infinity. The authors provides a $O(p^3)$ ($p$ is the number of features) algorithm to compute the regression coefficients. However, the method requires estimation of feature distribution for cases in the majority class. This is a daunting task in our scenario. Further research on methods to predict rare events in the presence of large and sparse features is required. Methods based on "shrinkage" estimation may prove useful here. However, the challenge is to scale them to massive datasets. Some recent work

that may be relevant includes techniques described in Ridgeway and Madigan (2002) and Huang and Gelman (2005). Yet another approach that has been pursued in the data mining community is that of scaling down the data using an approach called *data squashing* (Du Mouchel 2002; Du Mouchel and Agarwal 2003). Another approach that could be useful to reduce the dimension of feature space is clustering. However, the clustering here is done to maximize predictive accuracy of the model as opposed to a classical clustering approach that finds homogeneous sets in the feature space. It is also possible that clustering using an unsupervised approach may provide a good set of features for the prediction task and simplify the problem. The actual algorithms that are currently used by search engines are a complex combination of a number of methods.

## 1.5 ONLINE LEARNING

The discussion in the previous section pertains to estimating CTRs using retrospective data. Theoretically, if a model can predict CTR for all query/page, ad combination in different contexts, we are done. However, the number of queries/pages and ads is *astronomical*, making this infeasible in practice. Hence, one only ranks a subset of ads for a given query/page. The subset is decided based on some relevance criteria (e.g., consider only sports ads if the page is about sports). Thus, a large portion of query/page, ad space remains unexplored and may contain combinations that can lead to a significant increase in revenue. Also, the system is nonstationary and may change over time. Thus, ads that have been ruled out completely today in a given context might become lucrative after a month, but the retrospective estimation procedure would fail to discover them since it does not collect any data on such events. Designing efficient experiments to recover some of the lost opportunities is an important research problem that may lead to significant gains. Online learning or sequential design provides an attractive framework whereby a small fraction of traffic gets routed to the online learning system to conduct live experiments on a continuous basis. Although several online learning procedures exist, we will discuss the complexity of the problem and propose some potential solutions using a multi–armed bandit formulation.

We begin by providing a high-level overview of the multi–armed bandit problem and establish the connection to the CTR estimation problem in our context. In particular, we illustrate ideas using Content Match. The *multi–armed bandit problem* derives its name from an imagined slot machine with $k(\geq 2)$ arms. The $i$th arm has a payoff probability $p_i$ which is unknown. When arm $i$ is pulled, the player wins a unit reward with payoff probability $p_i$. The objective is to construct $N$ successive pulls of the slot machines to maximize the total expected reward. This gives rise to the familiar explore/exploit dilemma where, on the one hand, one would like to gather information on the unknown payoff probabilities, while on the other hand, one would like to sample arms with the best payoff probabilities empirically estimated so far. A bandit policy or allocation rule is an adaptive sampling process that provides a mechanism to select an arm at any given time instant based on all previous pulls and their outcomes. Readers lacking a background in statistics may ignore

the technical details in the next two paragraphs, but it will be insightful to understand the essential idea of the sampling process; the sampling scheme selects an arm that seems to have the potentail of getting the highest payoff at a given time instant. Thus, an arm with a worse empirical mean but high variance might be preferred to an arm with a better mean but low variance (exploration); after the sampling is continued for a while, we should learn enough to sample the arm that will provide the highest payoff (exploitation). A good sampling scheme should reach this point quickly. For instance, treating the ads that could be shown on a fixed webpage as arms of a bandit, an ad that has been shown on the page only twice and has received 1 click might be placed again on the page compared to an ad that had been shown 100 times and received 55 clicks.

A popular metric used to measure the performance of a policy is called *regret*, which is the difference between the expected reward obtained by playing the best arm and the expected reward given by the policy under consideration. A large body of bandit literature has considered the problem of constructing policies that achieve tight upper bounds on regret as a function of the time horizon $N$ (total number of pulls) for all possible values of the payoff probabilities. The seminal work of Lai and Robbins (1985) showed how to construct policies for which the regret is $O(\log N)$ asymptotically for all values of payoff probabilities. The authors further proved that the asymptotic lower bounds for the regret are also $\Omega(\log N)$ and constructed policies that actually attain them. Subsequent work has constructed policies that are simpler and achieve the logarithmic bound uniformly rather than asymptotically (see Auer et al. 2002 and references therein). The main idea in all these policies is to associate with each arm a priority function which is the sum of the current empirical payoff probability estimate plus a factor that depends on the estimated variability. Sampling the arm with the highest priority at any point in time, one explores arms with little information and exploits arms which are known to be good based on accumulated empirical evidence. With increasing $N$, the sampling variability is reduced and one ends up converging to the optimal arm. This clearly shows the importance of the result proved by Lai and Robbins (1985), which proves that one cannot construct the variance adjustment factor to make the regret better than $\Omega(\log N)$, thereby providing a benchmark for evaluating policies.

Two policies that both have $O(\log N)$ regret might involve different constants in the bounds (the constant depends on the *margin* of the bandit, which is the difference between the payoffs of the best two arms; the smaller the margin, the higher the constant) and may behave differently in real applications, especially when considering short-term behavior. One way of comparing the short-term behavior of policies that are otherwise optimal in the asymptotic sense is by using simulation experiments. One can also evaluate short-term behavior by proving the finite sample properties of policies, but this may become extremely hard to derive except in simple situations. The main difficulty is caused by the presence of dependencies in the sampling paths.

Focusing on Content Match for the sake of illustration, we can consider the online learning problem of matching ads to pages as a set of bandit processes. Thus, for each page, we have a bandit where ads are the arms and CTRs are the payoff probabilities. However, high dimensionality makes the bandit convergence slow

and involves a significant amount of exploration leading to revenue loss. In fact, asymptotic guarantees are not good enough in our situation, and we need procedures that can guarantee good short-term performance. Also, we need to learn the CTRs of the top few arms instead of the best arm, since we may run out of best ads due to budget constraints imposed by advertisers. Hence, given two policies that have similar revenue profiles, we would prefer the one whose CTR estimates have lower mean squared error.

To deal with the difficulties mentioned above, reducing dimensionality is of paramount importance. One approach is to assume that CTRs are simple functions of both page and ad features (Abe et al. 2003). Another approach is to cluster the pages and ads and conduct learning at coarser resolutions. Panoly et al. (2007) discuss such an approach where CTRs are learned at multiple resolutions, from coarser to finer, by using an online multistage sampling approach coupled with a Bayesian model. The authors report significant gains compared to a bandit policy that uses single-stage sampling. Further, they show that use of a Bayesian model leads to substantial reduction in mean square error without incurring loss in revenue. We note that sequential designs have been mainly considered in statistics in the context of clinical trials (see Rosenberger and Lachin 2002 for an overview). However, the problems in Internet advertising are large and require further research before sequential designs become an integral part of every ad network.

## 1.6 DISCOUNTING ADVERTISER TRAFFIC

The pay-per-click (PPC) revenue model used in Sponsored Search and Content Match is prone to abuse by unscrupulous sources. For instance, in Content Match, publishers who share the revenue proceeds from advertisers with the ad network might be tempted to use a service which uses sophisticated methods to produce false clicks for ads shown on the publisher's webpage. Although ad networks may benefit in the short term, collusion between publishers and ad networks is ruled out since such false clicks dilute the traffic quality received by advertisers through clicks on ads and lead to substantial losses to the ad network in the long run. Hence, monitoring traffic quality on the publisher's webpage is extremely important and, to a large extent, determines the feasibility of the PPC model in the long run. Ad networks have built sophisticated systems to detect such false clicks in order to protect their advertisers. In Sponsored Search, a competitor might use a similar behavior to drain a competitors' advertising budget. The problem, popularly known as *click fraud*, has received a lot of attention in recent times, including lead articles in *Business Week* and the *New York Times*. Another fraudulent behavior used in Sponsored Search is known as *impression fraud*. Here, an advertiser may use a robot to artificially inflate the impression volume and hence substantially deflate the CTR of competitors' ads. This, in turn, increases the rank of the advertiser's ads (ads are ranked using relevance measured by both CTR and bid) and increases his or her CTR. Thus, the advertiser gets better conversion rates at a lower cost.

The problems described above are difficult, and a complete solution seems to be elusive at this time. Simple frauds[2] intiated by a single individual manually (e.g., relatives of a blog owner clicking on ads, a person hired to click on ads of a competitor) are fairly obvious. Those that are initiated by more sophisticated means (e.g., randomizing false clicks over a large set of ips) are difficult to detect. An indirect approach is to use labels on good clicks to determine overall quality of clicks that are received on a publisher's website in Content Match and for an advertiser in Sponsored Search. Such labels can be obtained by tracking the behavior of users once they get to the landing page (the website of the advertiser) of the clicked ad. However, such data might be hard to obtain since advertisers are reluctant to allow the ad network to track the revenue generated through advertisements. Fortunately, some advertisers (not representative of the entire population) have agreed to share such data with the ad network, providing a valuable resource to validate automated algorithms built to detect false clicks. As more advertisers agree to provide such data, the situation will improve. The ideal approach here would be to have algorithms which can score every click as valid or invalid in an online fashion. However, this may be too ambitious, and an alternative approach which provides a global measure of click quality separately for advertisers and publishers based on a large pool of click data retrospectively may be a more feasible approach. Hybrid approaches that combine online and offline scoring may also be attractive.

Statistics has an important role to play here. One helpful approach is to detect abnormal click behavior in the highly multidimensional feature space that includes ip addresses, queries, advertisers, users (tracked by their browser cookie), ads and their associated features, and webpages and their associated features through time. This problem, known as *anomaly detection*, has received considerable attention in recent times in biosurveillance (Fienberg and Shmeuli 2005; Agarwal et al. 2006), telecommunications (Hill et al. 2006), monitoring help lines (Agarwal 2005), and numerous other areas. However, the percentage of anomalies in all the applications cited above is rare, which is typically not the case for click fraud. Popular press articles cite numbers ranging from 10% to 15% (although the distribution across several segments can vary widely). Time series methods to monitor the system over time (e.g., West and Harrison 1997) are germane in this context. Semisupervised learning approaches (sequentially labeling data to learn a classifier with a small set of labels but a large set of unlabeled examples) (Chapelle et al. 2006) are also important in this context. Not much research has been done in the statistics literature on semisupervised learning.

## 1.7 SOCIAL SEARCH

Internet advertising and search engines are a recent phenomenon, but they have had a profound impact on our lives. However, the current technology is constantly changing, and statisticians, computer scientists, machine learners, economists, and

[2]The term *fraud* is used loosely here; it means "unethical" in this context.

social scientists have an important role in shaping the next generation of search engines and Internet advertising. One important direction is social search. The popularity of Web-based tagging systems like Del.icio.us, Technocrati, and Flickr, which allow users to annotate resources like blogs, photographs, web pages, etc with freely chosen keywords (*tags*) (see Marlow et al. 2005 for an overview) has provided a rich source of data that can potentially be exploited to improve and broaden search quality, which will in turn increase ad revenue. These tagging systems also allow users to share their tags among friends. How does one exploit this rich source of information and the corresponding social network among users to enhance search quality? Let us consider the social bookmarking site Del.icio.us, for example. In Del.icio.us, users can *post* the URLs (called *artifacts*) of their favorite webpages into their Del.icio.us account and annotate these artifacts with informative tags. Users can also include their friends and other like-minded people in their social network. When searching for artifacts relevant to a particular keyword, it seems intuitive that apart from the relevance of content in artifacts to the keyword, one could further improve the relevance of search results by incorporating the tagging behavior of the user and others in his or her social network. For instance, a search for the keyword *conference* by the author should rank all statistics conference higher for the author, since most of his friends have bookmarked recent statistics conferences. A matching based on content alone might provide high rank to a conference on chemistry, which is perhaps not that interesting to the author. Incorporating user tags and the social network of users to personalize the search is a promising new area.

Currently, the search engine and publisher network monetizes its services through an ad network. Is it possible to build a network where individuals in a social network actively participate in providing answers to queries and enhancing the search? How would one design incentives to create a reasonable probability of extracting answers out of the network? Kleinberg and Raghavan (2005) explore theoretical properties of this fascinating idea.

## 1.8 CONCLUSION

In this chapter, we have provided an overview of Internet advertising and emphasized the important role statisticians can play as technology creators (as opposed to technology aiders) through a set of examples. The challenges discussed in this chapter are by no means exhaustive and provide a perspective based on the author's experience at a major search engine company for a period of one year. As a disclaimer, the views expressed are solely the author's own and are in no way representative of the official views of his employer. Although several statisticians have made a transition to this exciting area, more will be needed in the coming years. Internet advertising provides a unique opportunity to shape the future of the Web and invent technology that can affect the lives of millions of people. The author would like to urge statisticians to consider this area when making a career decision. One important component in conducting research in the area is the availability of data. Several search engine companies are trying their best to provide sanitized data for academic research.

However, the recent AOL debacle wherein search logs containing private information about users were released to the public demonstrates the difficulty of the problem. Hence, for research that depends critically on real data, the best method at the moment seems to involve working in close collaboration with companies that collect such data on an ongoing basis.

## ACKNOWLEDGMENTS

I thank Chris Olston and Arpita Ghosh for discussions and pointers to related work in crawling and auction theory. I also benefited from discussions with Srujana Merugu, Michael Benedikt, and Sihem Amer-Yahia on social search. I would also like to thank an anonymous referee and the editors, whose insightful comments improved the presentation of the chapter.

## REFERENCES

Abe, N., Biermann, A.W., and Long, P.M. (2003). Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4): 263–293.

Agarwal, D. (2005). An empirical bayes approach to detect anomalies in dynamic multidimensional arrays. *International Conference on Data Mining*.

Agarwal, D., McGregor, A., Phillips, J.M., Venkatasubramanian, S., and Zhu, Z. (2006). Spatial scan statistics: Approximations and performance study. *SIGKDD. Knowledge Discovery and Data Mining*.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47: 235–256.

Chapelle, O., Schlkopf, B., and Zien, A. (eds.). (2006). *Semi-supervised Learning*. Cambridge, MA: MIT Press.

Chawla, N., Japkowicz, N., and Kolcz, A. (eds.). (2003). Learning from Imbalanced Datasets. *Proceedings of the icml2003 Workshop*.

Chawla, N., Japkowicz, N., and Kolcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1): 1–6.

Cho, J. and Garcia-Molina, H. (2003). Estimating frequency of change. *ACM Transactions on Internet Technology*, 3(3): 256–290.

Cho, J. and Ntoulas, A. (2002). Effective change detection using sampling. *Very Large Databases*.

Dasgupta, A., Ghosh, A., Kumar, R., Olston, C., Pandey, S., and Tomkins, A. (2007). Discoverability of the web. *World Wide Web*.

DuMouchel, W. (2002). *Data Squashing: Constructing Summary Data Sets*. Norwell, MA: Kluwer Academic Publishers.

DuMouchel, W. and Agarwal, D. (2003). Applications of sampling and fractional factorial designs to model-free data squashing. *Knowledge Discovery and Data Mining*.

Edelman, B., Ostrovsky, M., and Schwarz, M. (2006). Internet advertising and the generalizaed second price auction: Selling billions of dollars worth of keywords. Second Workshop on Sponsored Search Auctions, Ann Arbor, Michigan, June.

Fienberg, S.E. and Shmueli, G. (2005). Statistical issues and challenges associated with rapid detection of bio-terrorist attacks. *Statistics in Medicine*, 24(4): 513–529.

Gittins, J.C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, 41: 148–177.

Hill, S., Agarwal, D., Bell, R., and Volinsky, C. (2006). Building an effective representation for dynamic graphs. *Journal of Computational and Graphical Statistics*, 15: 584–608.

Huang, Z. and Gelman, A. (2005). Sampling for bayesian computation with large datasets. Technical Report, Columbia University.

Japkowicz, N. (2000). Learning from imbalanced data sets: Papers from the aaai workshop. aaai, 2000. Technical Report WS-00-05,

King, G. and Zeng, L. (2001). Logistic regression in rare events data. *Political Analysis*, 9(2): 137–163.

Kleinberg, J.M. and Raghavan, P. (2005). Query incentive networks. In *FOCS '05: 46th Annual IEEE Symposium on Foundations of Computer Science*.

Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6: 4–22.

Manning, C.D., Raghavan, P., and Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2005). Position paper, tagging, taxonomy, flickr, article, toread. *WWW, Collaborative Web Tagging Workshop*.

Owen, A. (2007). Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, 8: 761–773.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). Pagerank citation ranking: Bringing order to the web. Technical Report, Stanford University.

Pandey, S., Agarwal, D., Chakrabarti, D., and Josifovski, V. (2007). Bandits for taxonomies: a model based approach. *Proceedings of the Siam Data Mining Conference*.

Ridgeway, G. and Madigan, D. (2002). A sequential monte carlo method for bayesian analysis of massive datasets. *Journal of Data Mining and Knowledge Discovery*, 7: 301–319.

Rosenberger, W.F. and Lachin, J.M. (2002). *Randomization in Clinical Trials: Theory and Practice*. New York: Wiley.

Varian, H.R. (2007). Position auctions. *International Journal of Industrial Organization*, 25: 1163–1178.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer-Verlag.