International Journal of Electrical and Computer Engineering (IJECE)
Vol. 4, No. 5, October 2014, pp. 810~816
ISSN: 2088-8708

Issues and Challenges in Advertising on the Web

Deepthi Gurram, B. Vijaya Babu, Vidyullatha Pellakuri
CSE Department, KL University, Vaddeswaram, Guntur, Andhra Pradesh, India.

Article Info

Article history:

Received Aug 1, 2014
Revised Sep 2, 2014
Accepted Sep 20, 2014

Keyword:

Adwords
Greedy algorithm
Matchin
Off-line
On-line

ABSTRACT

One of the big surprises of the 21st century has been the ability of all sorts of interesting Web applications to support themselves through advertising, rather than subscription. While radio and television have managed to use advertising as their primary revenue source, most media – newspapers and magazines, for example – have had to use a hybrid approach, combining revenue from advertising and subscriptions. A venue for on-line advertising has been search, and much of the effectiveness of search advertising came from the "adwords" model of matching search queries to advertisements. This paper presents the algorithms for optimizing the way of matching search queries to advertisements is done. The algorithms discussed are of unusual type; they are greedy and they are on-line which are used to tackle the adwords problem.

Corresponding Author:

Deepthi Gurram,
Departement of Computer Science & Engineering,
KL University,
Vaddeswaram, Guntur (Dt), Andhra Pradesh, India.
Email: deepug2110@gmail.com

1. INTRODUCTION

The Web offers many ways for an advertiser to show their ads to potential customers. Some sites, such as eBay, Craig's List or auto trading sites allow advertisers to post their ads directly, either for free, for a fee, or a commission. Advertisers pay for the display at a fixed rate per impression (one display of the ad with the download of the page by some user). Normally, a second download of the page, even by the same user, will result in the display of a different ad and is a second impression. Search ads are placed among the results of a search query. Advertisers bid [10] for the right to have their ad shown in response to certain queries, but they pay only if the ad is clicked on. The particular ads to be shown are selected by a complex process, involving the search terms that the advertiser has bid for, the amount of their bid, the observed probability that the ad will be clicked on, and the total budget that the advertiser has offered for the service [5].

When advertisers can place ads directly, such as a free ad on Craig's List or the "buy it now" feature at eBay, there are several problems that the site must deal with. Ads are displayed in response to query terms, e.g., "apartment Palo Alto." The Web site can use an inverted index of words, just as a search engine does and return those ads that contain all the words in the query. Alternatively, one can ask the advertiser to specify parameters of the ad, which are stored in a database [10]. For instance, an ad for a used car could specify the manufacturer, model, color, and year from pull-down menus, so only clearly understood terms can be used. Queryers can use the same menus of terms in their queries.

1.1 Web ad Placement

There are some vital factors for web ad placement that we state below:

- Positioning. The position of the ad on a page greatly influences its click ability. The click probability decreases exponentially as the position/sequence increases.

- Similarity to the user's search queries. The ad attractiveness correlates with the query terms.
- Specialized media (topic oriented magazines/ blogs with a high ads click-through rate). Statistics prove that users do prefer a web page relevant ad rather than unrelated messy stuff.
- User relevant. The web media use the advantage of having some information about the user. This enables them to choose the 'right' ad for 'the user'. Statistically and cumulatively one may determine a visitor's interest for certain things, for example:
  - o search queries (correlates with point 2)
  - o social nets and related groups
  - o email content exchange (being used extensively by Google - ethicalness is not quite clear)
  - o Bookmarks and back links

1.2 Issues for Display Ads

This form of advertising on the Web most resembles advertising in traditional media. An ad for a Chevrolet run in the pages of the New York Times is a display ad, and its effectiveness is limited. It may be seen by many people, but most of them are not interested in buying a car, just bought a car, don't drive, or have another good reason to ignore the ad [8] .Yet the cost of printing the ad was still borne by the newspaper and hence by the advertiser. An impression of a similar ad on the Yahoo! home page [9] is going to be relatively ineffective for essentially the same reason. The fee for placing such an ad is typically a fraction of a cent per impression. An ad for golf clubs on sports.yahoo.com/golf has much more value per impression than does the same ad on the Yahoo! home page or an ad for Chevrolets on the Yahoo! golf page [9]. However, the web offers an opportunity to display ads in such a way that it is possible to use information about the user to determine which ad they should be shown, regardless of what page they are looking at. Here raises the issues of privacy.

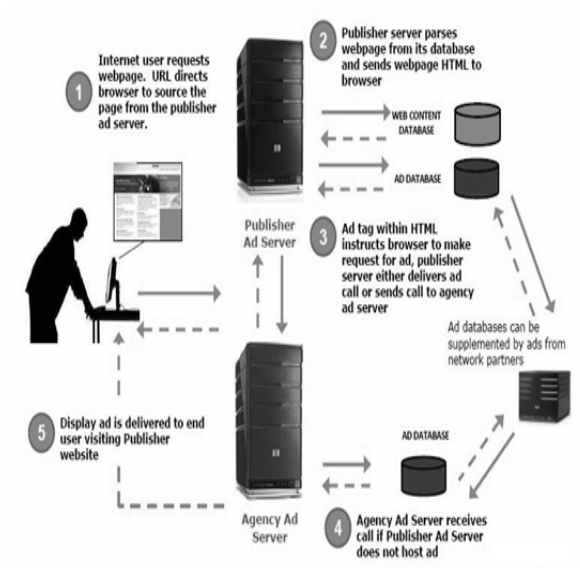

Figure 1. Workflow of displaying advertisements

The challenge of effective web advertisement primarily involves placing relevant ads on user requested web pages. Those ads must be relevant to a page receiver that is relevant to the page context and/or directly to the user [10].

2. RELATED WORK

What algorithms are to be applied in placing ads in modern web business? The algorithms for getting ads, that are relevant to search queries, must be of the on-line nature. Strictly, on-line algorithms are those where data input is fed into the algorithm, without having the entire input available from the start. It very much looks like signal processing (DSP), where the input stream is not evident from the start. Input data stream might also be so fast, that the processing latency is of much higher importance compared to static presented data input.

'On-line' here refers to the nature of the algorithm, and should not be confused with 'on-line' meaning 'on the Internet' [10].

Using off-line algorithms for ad placement, we just observe search queries for a time period, and consider the bids advertisers made on search terms, as well as their advertising budgets for the time period. Thus the algorithm can then assign ads to the queries in a way that maximizes both the revenue to the search engine and the number of impressions that each advertiser gets. But since users who issue queries can't wait for the aggregate results, on-line algorithms enter in here.

With on-line algorithms we may use past data for computing (like click-through rate), yet we cannot assume having certain queries or events in the future.

2.1 On-Line Algorithms

Matching advertisements to search queries is referred to as "on-line", and they generally involve an approach called"greedy". A preliminary example of an on-line greedy algorithm for a simpler problem: maximal matching is shown [10].

2.2 On-Line and Off-Line Algorithms

The work flow of the typical algorithms is as follows. All the data needed by the algorithm is presented initially. The algorithm can access the data in any order. At the end, the algorithm produces its answer. Such an algorithm is called off-line.

There is an extreme form of stream processing, where we must respond with an output after each stream element arrives. We thus must decide about each stream element knowing nothing at all of the future. Algorithms of this class are called on-line algorithms [4]. As the case in point, selecting ads to show with search queries would be relatively simple if we could do it off-line. We would see a month's worth of search queries, and look at the bids advertisers made on search terms, as well as their advertising budgets for the month, and we could then assign ads to the queries in a way that maximized both the revenue to the search engine and the number of impressions that each advertiser got. The problem with off-line algorithms is that most queryers don't want to wait a month to get their search results.

Thus, we must use an on-line algorithm to assign ads to search queries. That is, when a search query arrives, we must select the ads to show with that query immediately. We can use information about the past, e.g., we do not have to show an ad if the advertiser's budget has already been spent, and we can examine the click-through rate (fraction of the time the ad is clicked on when it is displayed) that an ad has obtained so far.

3. GREEDY ALGORITHMS

Many on-line algorithms are of the greedy algorithm type. These algorithms make their decision in response to each input element by maximizing some function of the input element and the past.

Example 1: A manufacturer A of replica antique furniture has bid 10 cents on the search term "chesterfield". A more conventional manufacturer B has bid 20 cents on both the terms "chesterfield" and "sofa." Both have monthly budgets of $100, and there are no other bidders on either of these terms. It is the beginning of the month, and a search query "chesterfield" has just arrived [10].

For the data of this example, what will happen is that the first 500 "sofa" or "chesterfield" queries will be assigned to B. At that time, B runs out of budget and is assigned no more queries. After that, the next 1000 "chesterfield" queries are assigned to A, and "sofa" queries get no ad and therefore earn the search engine no money. The worst thing that can happen is that 500 "chesterfield" queries arrive, followed by 500 "sofa" queries. An off-line algorithm could optimally assign the first 500 to A, earning $50, and the next 500 to B, earning $100, or a total of $150. However, the greedy algorithm will assign the first 500 to B, earning $100, and then has no ad for the next 500, earning nothing.

3.1 The Matching Problem

A problem is discussed which is a simplified version of the problem of matching [2] ads to search queries. This problem, called "maximal matching," is an abstract problem involving bipartite graphs (graphs with two sets of nodes – left and right – with all edges connecting a node in the left set to a node in the right set).

3.2 Matches and Perfect Matches

Suppose we are given a bipartite graph [3] [6]. A matching is a subset of the edges such that no node is an end of two or more edges. A matching is said to be perfect if every node appears in the matching. Note that a matching can only be perfect if the left and right sets are of the same size. A matching that is as large as any other matching for the graph in question is said to be maximal.

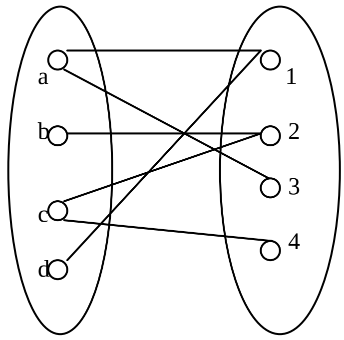

Figure 2. A Bipartite graph

The set of edges {(a, 1), (b, 2), (c, 4)} is a matching for the bipartite graph [3] [6] of Figure 2. Each member of the set is an edge of the bipartite graph, and no node appears more than once. The set of edges {(a, 3), (b, 2), (c, 4), (d, 1)} is a perfect matching, represented by heavy lines in Figure 3. Every node appears exactly once. There is a sole perfect matching for this graph, whereas some bipartite graphs have more than one perfect matching [2]. The matching of Figure 3 is also maximal, since every perfect matching is maximal.

3.3 The Greedy Algorithm for Maximal Matching

In particular, the greedy algorithm [7] for maximal matching works as follows. We consider the edges in whatever order they are given. When we consider (x, y), add this edge to the matching if neither x nor y are ends of any edge selected for the matching so far. Otherwise, skip (x, y).

Let us consider a greedy match for the graph of Figure 2. Suppose we order the nodes lexicographically, that is, by order of their left node, breaking ties by the right node. Then we consider the edges in the order(a,1),(a,3),(b,2),(c,2),(c,4),(d,1). The first edge, (a, 1), becomes part of the matching. The second edge, (a, 3), cannot be chosen, because node a already appears in the matching. The third edge, (b, 2), is selected, because neither node b nor node 2 appears in the matching so far. Edge (c, 2) is rejected for the match because 2 is already matched, but then (c, 4) is added to the match because neither c nor 4 has been matched so far. Finally, (d, 1) is rejected because 1 appears in the match. Thus, the matching produced by the greedy algorithm for this ordering of the edges is {(a, 1), (b, 2), (c, 4)}. This matching is not maximal.

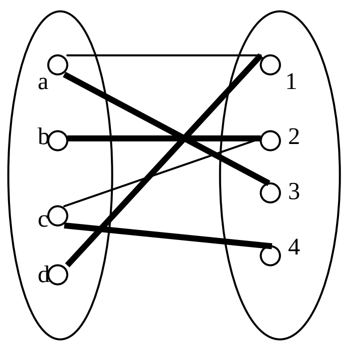

Figure 3. The only perfect matching for the graph of Figure 2

4. THE ADWORDS PROBLEM

The fundamental problem of search advertising, which we term the “adwords problem,” because it was first encountered in the Google Adwords system [1] [10]. Of course, the decision regarding which ads to show must be made on-line. The on-line algorithms for solving the adwords problem are as follows.

• Given:
1. A set of bids by advertisers for search queries.
2. A click-through rate for each advertiser-query pair.
3. A budget for each advertiser. We shall assume budgets are for a month, although any unit of time could be used.
4. A limit on the number of ads to be displayed with each search query.

• Respond to each search query with a set of advertisers such that:
1. The size of the set is no larger than the limit on the number of ads per query.
2. Each advertiser has bid on the search query.
3. Each advertiser has enough budgets left to pay for the ad if it is clicked upon.

Much of the effectiveness of search advertising comes from the “adwords” model of matching search queries to advertisements. Search ads are placed among the results of a search query. Advertisers bid for the right to have their ad shown in response to certain queries, but they pay only if the ad is clicked on. The particular ads to be shown are selected by a complex process, which we consider in this post. It embraces search terms that the advertiser has bid for, the amount of their bid, the statistical probability that the ad will be clicked on, and the total budget the advertiser has offered for the service.

4.1 The Greedy Approach to the Adwords Problem

Since only an on-line algorithm is suitable for the adwords [1] problem, we should first examine the performance of the obvious greedy algorithm. Suppose there are two advertisers A and B, and only two possible queries, x and y. Advertiser A bids only on x, while B bids on both x and y. The budget for each advertiser is 2. Let the sequence of queries be xxyy. The greedy algorithm is able to allocate the first two x’s to B, whereupon there is no one with an unexpended budget to pay for the two y’s. The revenue for the greedy algorithm in this case is thus 2. However, the optimum off-line algorithm will allocate the x’s to A and the y’s to B, achieving revenue of 4[10].

4.2 Competitive Ratio

Competitive ratio is the minimum rate of the online algorithm compared to the optimum off-line algorithms for the same case/problem over all possible inputs (especially the worst case). c = efficiency (online)/efficiency (offline) in the worst case. Through some calculation one may prove that generally for the greedy algorithm [7] in ads problem the competitive ratio is not less than 1/2.

4.3 Balance Algorithm

But there are other than greedy algorithms, giving better a competitive ratio. Besides the maximum present revenue balance, the balance algorithm considers also the greatest remaining budget of an advertiser in order to decide which ad to show. Its ultimate competitive ratio (simplified adwords model) will go close to 0.63 as the number of bidders and queries grow. However it does not work optimally in some extreme cases.

Example: In this example we compare the simplified situation for off-line and greedy on-line algorithms in ad placement.

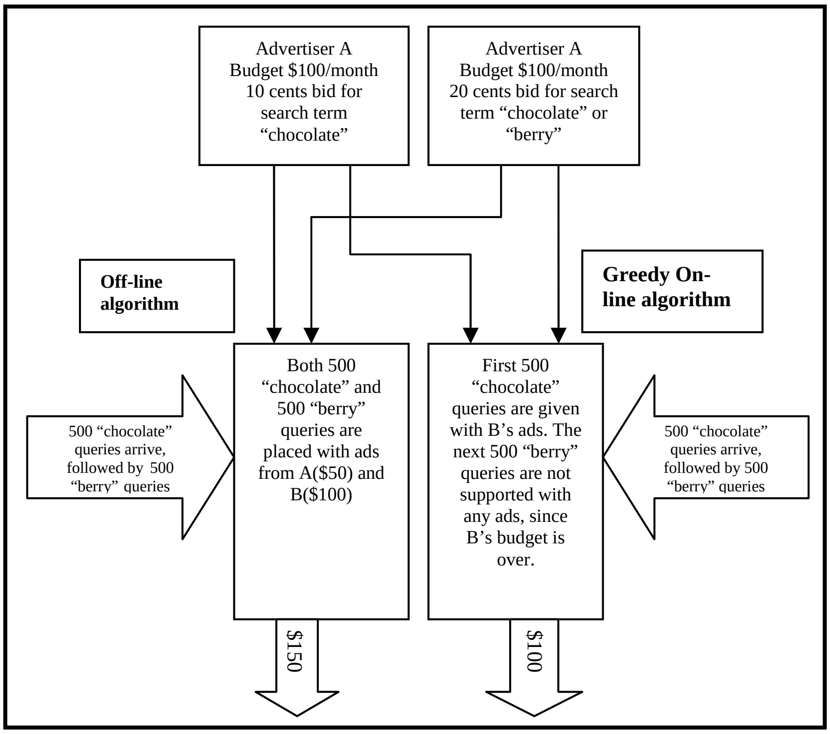

Figure 4. Example of comparison of off-line and greedy on-line algorithms.

Advertiser 'A' has bid 10 cents on the search term "chocolate". Advertiser B has bid 20 cents on both the terms "chocolate" and "berry". Both have monthly budgets of $100. We have no past statistics on either of these terms. We presuppose that only one ad is to be too displayed with one query. The obvious thing to do is to display B's ad, because they bid more (greedy approach). However, suppose there will be lots of search queries this month for "berry" but very few for "chocolate." Then A will never spend its $100 budget, while B will spend its full budget even if we give the query to A.

The worst case that can happen is that 500 "berry" queries arrive, followed by 500 "chocolate" queries (see the figure below, at right). The greedy algorithm will assign the first 500 ads to B, earning $100, and then has no ad for the next 500, earning nothing. The competitive ratio in this case is equal 2/3.

5. CONCLUSION

These basic theses and the simplified example have shown the challenge and also an elementary approach (greedy, on-line) to ad placement related to search queries. The data mining algorithms for ad placement in response to a search query or larger documents (emails) are on-line greedy or generalized balance algorithms on the match graphs. The complex query/bid matching process is to be done with hashing sets-of-words tables with the consequent matching of rare to un rare word order in order to find the total match of the bid set in the document.

REFERENCES

[1] A Mehta, A Saberi, U Vazirani, and V Vazirani. "Adwords and generalized on-line matching". IEEE Symp. on Foundations of Computer Science. 2005: 264–273.
[2] Bala Kalyanasundaram and Kirk R. Pruhs. An optimal deterministic algorithm for online b-matching. Theor. Comput. Sci., 2000; 233(1-2):319-325.
[3] RM Karp, UV Vazirani, and VV Vazirani. *An optimal algorithm for on-line bipartite matching*. In STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing. 1990: 352-358.
[4] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*. 2007; 54(5): 22.
[5] P Rusmevichientong and DP Williamson. An adaptive algorithm for selecting profitable keywords for search based advertising services. In *Seventh ACM Conference on Electronic Commerce*. 2006.

[6] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *STOC*. 2011.
[7] Hochbaum DS, A Pathria. "Analysis of the greedy approach in problems of maximum k-coverage". *Naval Research Logistics*. 1998; 45: 615-627.
[8] Revenue Science http://www.revenuescience.com/advertisers/advertiser_solutions.asp
[9] Yahoo! Smart Ads http://advertising.yahoo.com/marketing/smartads/
[10] Anand Rajaraman, Jeffrey David Ullman. "Mining of Massive Datasets".