Adword Auctions:

How internet search engine companies such as google, Yahoo and MSN, decide what ads to display with each query so as to maximize its revenue (profit maximization).

customers typing in keywords, called Adwords by google

The Adwords market is essentially a large auction where business place bids for individual keywords (Adwords) together with limits specifying their maximum daily budget. The search engine company earns revenue from business when it displays theirs ads in response to a relevant search query (if the user actually clicks on the ad and also it charges the second price, but we ignore these issues for now)

It turns out the problem is a generalization of the online bipartite matching problem. and there is a competitive ratio $1-\frac{1}{e}$ for some cases of this problem (as we see later). Note that since the Adwords are coming one by one the problem is an online Auction. is some sense.

2

Notes for Online Auction

Koyel Mukherjee

October 6, 2010

Search engines like Google and Yahoo! conduct online ad auctions and this generates a huge amount of revenue for them. While searching, users enter keywords, and for certain keywords, certain commercial advertisers are interested in showing up in the search results. It is possible that the user might be interested in commercial search results for that keyword they have entered. If the user actually goes and clicks on this ad, and buys the product, the advertiser gains money. Even just by showing up in the search results, the advertiser stands to gain, as it is good advertisement to the user.

Search engines conduct online auctions for these keywords among the interested advertisers, and they want to maximize their revenue. When a keyword comes in, the ad slots to be shown up in the search results are instantly sold to interested advertisers, by conducting an auction. Since the keywords are not known in advance, and as and when a keyword comes in, the bids by different advertisers get revealed, the nature of the auction is online. We assume a completely adversarial setting, by which we mean, we make no assumption on the input pattern of keywords. Also, ad slots need to be sold instantly as and when a keyword arrives, and the decision is irrevocable. We also assume in this talk, that there is only one ad slot that needs to be auctioned, and also, the advertisers bid their true value. That is, ensuring truthfulness is not our objective in this talk. However, the advertisers have a fixed budget, and we cannot exceed the budget in our allocations.

In this talk we outline the procedure given by Buchbinder, Jain and Naor [1]. They give a primal dual mechanism achieving a competitive ratio of $1-\frac{1}{e}$ asymptotically, matching the ratio given earlier by Mehta et al. [3]. However, the analysis presented by Buchbinder et al. is easier to understand, and does not use a tailor-made potential function for analysis, as used by Mehta et al. One crucial assumption in both these works is that the individual bid is small compared to the budget of any advertiser, in other words, the budgets are very large.

The underlying setting can be thought of as a bipartite graph. One side of the bipartition consists of the nodes corresponding to the advertisers. These set of vertices are known to the system from the start. Let this set of advertisers be I, and $|I| = n$. The other side of the bipartition consists of nodes corresponding

(3)

to the keywords, and these nodes along with the edges incident on them get revealed in an online manner, when they arrive. Let the set of keywords be J, and $|J| = m$. Each advertiser has a daily budget B_i, known to the system. We want to assign each online arriving keyword node to one interested advertiser, who has an edge to this keyword node, and we want to maximize the weight of the matching. Once we decide on including an edge in the matching, we cannot change our decision. However, we should not exceed the budget of the advertiser. The total money that the advertiser will pay us is his budget, even if he bids in excess of his budget. If all the bids were restricted to be 0 or 1 and the budgets very large $\to \infty$, then this becomes the online b-matching problem, for which Kalyanasundaram and Pruhs [2] had proved that the best competitive ratio one can get is $1 - \frac{1}{e}$, and they gave an algorithm BALANCE that achieves this competitive ratio. Mehta et al. [3] proved that $1 - \frac{1}{e}$ is tight even for the online auctions case, and present an algorithm that asymptotically achieves this ratio, when budgets are very large. Buchbinder et al. also give an algorithm with the same competitive ratio achieved asymptotically, but with a cleaner primal-dual analysis.

The offline primal problem is the following:

$$\max \sum_i \sum_j b_{ij} y_{ij} \ s.t.: \ \sum_i y_{ij} \leq 1 \, \forall j \in J, \ \sum_j b_{ij} y_{ij} \leq B_i \, \forall i \in I, \ y_{ij} \geq 0 \, \forall i \in I, j \in J \tag{1}$$

This was proved to be NP-hard. The dual for the above problem is

$$\min \sum_i B_i x_i + \sum_j z_j \ s.t.: \ b_{ij} x_i + z_j \geq b_{ij}, \ x_i, z_j \geq 0 \, \forall i \in I, \ \forall j \in J \tag{2}$$

We want to use weak duality to bound the competitive ratio. Since the primal problem is maximization, by weak duality, any feasible primal solution P and any feasible dual solution D is related as $P \leq D$. This is true even for the primal optimum solution, OPT. Therefore $D \geq OPT$. We want start from 0 value both the primal and dual solutions, and at every round of online auction(keyword arrival), we increment our primal solution by ΔP and dual solution by ΔD. At the end our primal is $\sum \Delta P$ and the dual is $\sum \Delta D$. We would try to bound the ratio of $\Delta P / \Delta D$ at every round. Since the ratio of $1 - \frac{1}{e}$ is tight for this problem, if we can bound $\Delta P / \Delta D \geq 1 - \frac{1}{e}$ at every round, then at the end our primal solution is $P \geq (1 - \frac{1}{e}) D \geq (1 - \frac{1}{e}) OPT$, and we get an algorithm with a competitive ratio $1 - \frac{1}{e}$.

One might wonder why we can't use greedy in this problem and just allocate the keyword to the highest bidder, but due to the budget constraint we can construct an example where greedy can only do as well as 1/2. Say, there are two types of keyword, w1 and w2, and two bidders each with a budget of N. Bidder 1 has bids 1 for both w1 and w2, whereas bidder 2 has bid 1 only for w1 and 0 for

4

w2. If now w1 comes N times, followed by w2 N times, we allocate greedily w1 to bidder 1 for all the N arrivals, and exhaust his budget. When w2 comes N times, we have no one to allocate. So we get a revenue of N, whereas OPT could have got 2N. Hence the intuition is to balance the allocation somehow. We should not allocate to the same bidder again and again. Mehta et al. [3] had a similar idea, and used a potential function to decrease the effective bid of an advertiser, depending on the how much his budget is already exhausted. Here, we outline the approach of Buchbinder et al. [1]. In this approach we assume only one keyword arrives at a time.

The algorithm is as follows:

1. Initially $\forall i, x_i \leftarrow 0$. (Implicitly all z_j and all y_{ij} at 0 to maintain the initial $P = 0$ and $D = 0$.
2. Upon arrival of a new keyword j, allocate to the advertiser $i \in argmax_{\hat{i} \in I} b_{\hat{i}j}(1 - x_{\hat{i}})$
3. If $x_i \geq 1$, do nothing. Otherwise
4. Charge i the minimum of b_{ij} and his budget, and set $y_{ij} \leftarrow 1$.
5. $z_j \leftarrow b_{ij}(1 - x_i)$ (explicitly modifying z_j only the one time it arrives. Each arrival is considered new.)
6. $x_i \leftarrow x_i(1 + \frac{b_{ij}}{B_i}) + \frac{b_{ij}}{(c-1)B_i}$, where $c = (1 + R_{max})^{\frac{1}{R_{max}}}$, where $R_{max} = \max_{i \in I, j \in J} \frac{b_{ij}}{B_i}$

Theorem: The algorithm is $(1 - \frac{1}{c})(1 - R_{max})$ competitive for the online budgeted allocation problem. The competitive ratio $\to \infty$ as $R_{max} \to 0$, in other words, as the budgets $B_i \to \infty \forall i \in I$ and the bids are small compared to budgets.

We will prove the theorem by proving the following three claims.

Claim 1: The algorithm produces a dual feasible solution.
$x_i \geq 0$ by assignment, since we only increment x_i, if we change it. Also, $z_j \geq 0$ by assignment. We update the value of z_j to $b_{ij}(1 - x_i)$, only if $x_i < 1$, and hence $(1 - x_i) > 0$. If $x_i \geq 1$, then the dual constraint $b_{ij}x_i + z_j \geq b_{ij}$ is automatically satisfied. For the advertisers for which $x_i < 1$, we update z_j to $\max_{\hat{i} \in I} b_{\hat{i}j}(1 - x_{\hat{i}})$.

$$z_j = b_{ij}(1 - x_i) \geq b_{\hat{i}j}(1 - x_{\hat{i}}) \forall \hat{i} \in I; \tag{3}$$

$$z_j + b_{\hat{i}j}x_{\hat{i}} \geq b_{\hat{i}j} \forall \hat{i} \in I \tag{4}$$

Claim 2: $\Delta P \geq \Delta D(1 - \frac{1}{c})$ in every iteration when there is a non-zero increment in ΔD and ΔP When there is a non-zero increment in the dual, we have:

$$\Delta D = B_i \Delta x_i + z_j = B_i(\frac{b_{ij}x_i}{B_i} + \frac{b_{ij}}{(c-1)B_i}) = b_{ij}(1 + \frac{1}{c-1}). \tag{5}$$

The increment in primal is b_{ij}, (because we set $y_{ij} = 1$ even if the remaining budget is less than the current bid. Hence $\Delta P / \Delta D = 1 - \frac{1}{c}$.

Claim 3: The algorithm produces an almost feasible primal solution.

$y_{ij} \geq 0$ always. However, the infeasibility may arise due to violation of budget, when the remaining budget is less than the bid, and we still set $y_{ij} = 1$. In such cases, $\sum_j b_{ij} y_{ij} \geq B_i$. Ideally we want x_i to become 1 when the budget is just exhausted, but because of arbitrary values of bids, it is difficult to ensure that. Instead, we prove a weaker claim: When $\sum_j b_{ij} y_{ij} \geq B_i$, then $x_i \geq 1$. This will ensure, that the budget may be violated in at most one iteration for every bidder. We show this by proving inductively:

$$x_i \geq \frac{c^{\frac{\sum_j b_{ij} y_{ij}}{B_i}} - 1}{c - 1}. \tag{6}$$

Hence, when the budget gets exhausted, $x_i \geq 1$. Initially it is true trivially. Let us assume it holds for bidder i at some interation k, when bidder i is chosen.

$$x_i(end) = x_i(start)(1 + \frac{b_{ik}}{B_i}) + \frac{b_{ik}}{(c-1)B_i}; \tag{7}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}} - 1}{c - 1}(1 + \frac{b_{ik}}{B_i}) + \frac{b_{ik}}{(c-1)B_i} \tag{8}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}}(1 + \frac{b_{ik}}{B_i}) - 1}{c - 1} \tag{9}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}} c^{\frac{b_{ik}}{B_i}} - 1}{c - 1} \tag{10}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_j b_{ij} y_{ij}}{B_i}} - 1}{c - 1} \tag{11}$$

where the first inequality follows from induction hypothesis, and the second one from the fact: $for 0 < x \leq y \leq 1, \frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$. (easy to prove). In this inequality, if we replace x with $\frac{b_{ik}}{B_i}$ and y by R_{max}, then we get,

$$\frac{\ln(1 + \frac{b_{ik}}{B_i})}{\frac{b_{ik}}{B_i}} \geq \frac{\ln(1 + R_{max})}{R_{max}} \tag{12}$$

But $\ln(1 + R_{max}) = \ln(c^{R_{max}}) = R_{max} \ln(c)$. It is easy to get the rest.

However, we still might have violated the budget in at most one iteration for every $i \in I$. The maximum violation is: $\sum_j b_{ij} y_{ij} \leq B_i + \max_{j \in J} b_i j$. Therefore,

we can lower bound the real (feasible) primal profit obtained by the algorithm as $\sum_j b_{ij} y_{ij} \frac{B_i}{B_i + \max_{j \in J} b_{ij}} \geq \sum_j b_{ij} y_{ij} \frac{1}{1+R_{max}} \geq \sum_j b_{ij} y_{ij} (1 - R_{max})$. The first inequality is by the definition of R_{max} and the second inequality is by binomial expansion.
Putting the three claims together, we have that our algorithm gives a feasible primal value $P_f \geq P(1 - R_{max}) \geq D(1 - \frac{1}{c})(1 - R_{max}) \geq (1 - \frac{1}{c})(1 - R_{max}) OPT$, at the end, thereby proving the theorem.

References

[1] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. *Algorithms–ESA 2007*, pages 253–264, 2007.

[2] B. Kalyanasundaram and K. Pruhs. An optimal deterministic algorithm for online b-matching. In *Foundations of Software Technology and Theoretical Computer Science*, pages 193–199. Springer, 1996.

[3] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.