Web Advertising
Part 2

Thanks for source slides and material to: J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

History of Web Advertising

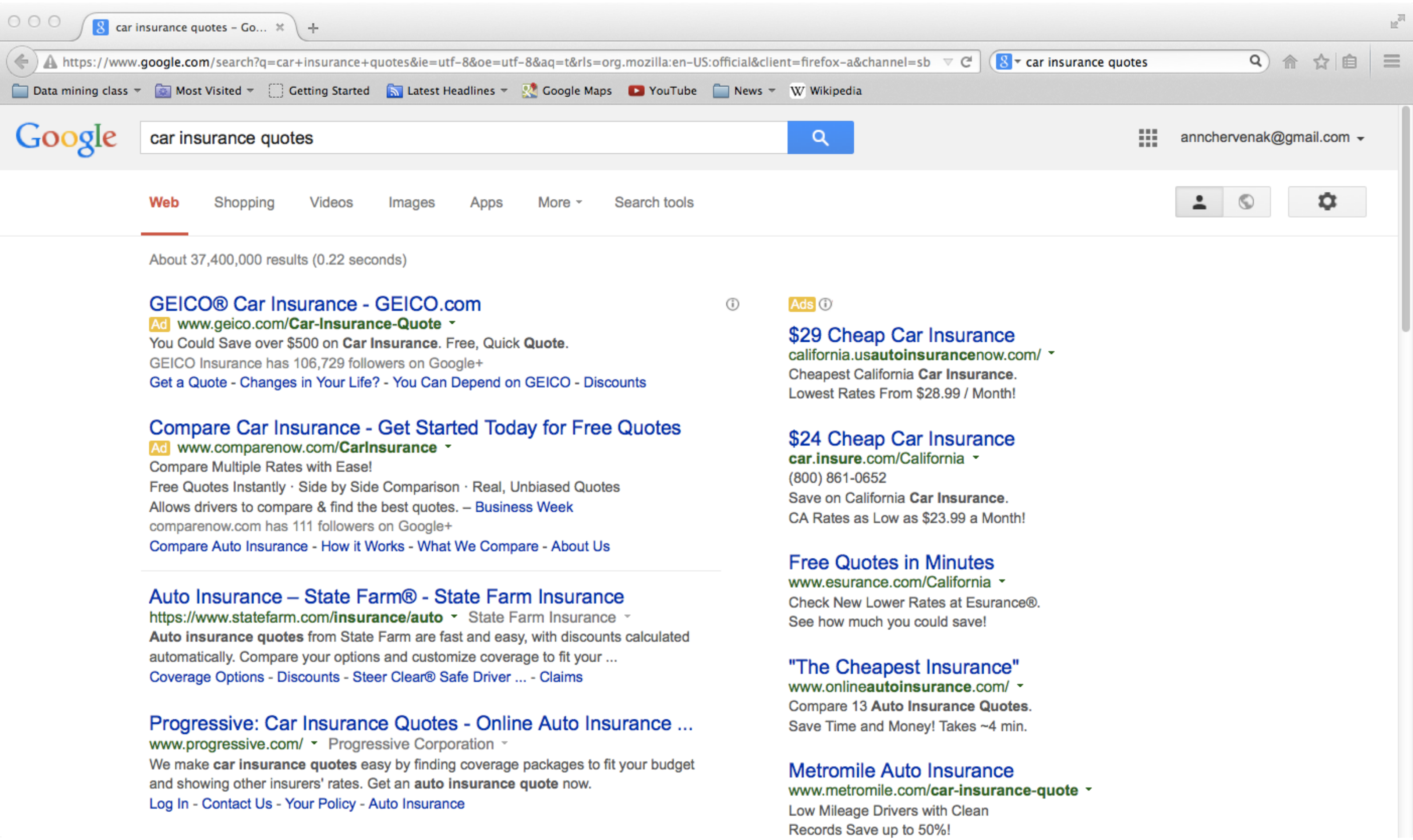
- **Banner ads (1995-2001)**
  - Initial form of web advertising
  - Popular websites charged *X*$ for every 1,000 "impressions" of the ad
    - Called "**CPM**" rate (Cost per thousand impressions)
    - Modeled similar to TV, magazine ads
  - From **untargeted** to **demographically targeted**
  - **Low click-through rates**
    - Low Return on Investment (ROI) for advertisers.

CPM…cost per *mille*
Mille…thousand in Latin

Performance-based Advertising

- **Introduced by Overture around 2000**
  - Advertisers **bid** on **search keywords**
  - When someone searches for that keyword, the **highest bidder's ad is shown first**
  - Advertiser is charged only if the ad is clicked on
- Similar model adopted by Google with some changes around 2002
  - Called **Adwords.**

Ads vs. Search Results



Web 2.0

- **Performance-based advertising works!**
  - Multi-billion-dollar industry
- **Interesting problem:**
  What ads to show for a given query?
  - (Today's lecture)
- **If I am an advertiser, which search terms should I bid on and how much should I bid?**
  - (Not focus of today's lecture).

Adwords Problem

- **Given:**
  - **1.** A set of bids by advertisers for search queries
  - **2.** A click-through rate (CTR) for each advertiser-query pair
  - **3.** A budget for each advertiser (say for 1 month)
  - **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 budget left to pay for the ad if it is clicked upon.

Adwords Problem

- A stream of queries arrives at the search engine: $q_1, q_2, \ldots$
- Several advertisers bid on each query
- When query q_i arrives, search engine must pick a subset of advertisers whose ads are shown

- **Goal: Maximize search engine's revenues**
  - **Simple solution:** Instead of raw bids, use the "**expected revenue per click**" (i.e., **Bid*CTR**)
- **Clearly we need an online algorithm!**

Invalid Clicks

- Thanks to:
  - Shweta Chandramouli
  - Siran Li
- More links:
  - http://www.google.com/ads/adtrafficquality/#click-fraud-and-invalid-traffic http://googleblog.blogspot.com/2008/03/using-data-to-help-prevent-fraud.html http://www.google.com/ads/adtrafficquality/invalid-click-protection.html
  - https://support.google.com/adwords/answer/2549113
  - https://support.google.com/adwords/answer/2375444
  - http://xcitemediagroup.com/competitors-clicking-my-ads-adwords-invalid-clicks
  - http://www.google.com/ads/adtrafficquality/index.html
  - https://support.google.com/adwords/answer/42995?hl=en

Adwords: Invalid Clicks

- https://support.google.com/adwords/answer/2375444#invalid
- **Invalid clicks** are basically clicks that Google doesn't consider "real" clicks,
  - such as **clicks made by a robot or automated clicking tools**.
- You can see the number and percentage of clicks that have been classified as invalid and automatically filtered from your account by adding the "Invalid clicks" column on your **Campaigns** or **Dimensions** tabs.
  - Don't worry, **you aren't charged for these clicks, and they don't affect your account statistics.**

More from Google…

- Here are just a few examples of what Google may consider to be invalid clicks:
  - **manual clicks intended to increase your advertising costs** or to **increase profits for website owners hosting your ads**
  - **clicks by automated clicking tools, robots, or other deceptive software**
  - **extraneous clicks that provide no value** to the advertiser, such as the second click of a double-click.

More from Google…

- **Each click on an AdWords ad is examined by our system**, and Google has sophisticated systems to identify invalid clicks and impressions and remove them from your account data.
- When Google determines that clicks are invalid, we try to **automatically filter them from your reports and payments** so that you're not charged for those clicks. If we find that invalid clicks have escaped automatic detection, you may be eligible to receive a credit for those clicks. These credits are called "invalid activity" adjustments.

Using data to help prevent fraud

- http://googleblog.blogspot.com/2008/03/using-data-to-help-prevent-fraud.html
- Our Ad Traffic Quality team: three-stage system for detecting invalid clicks.
- The three stages are: **(1) proactive real-time filters**, **(2) proactive offline analysis**, and **(3) reactive investigations**.
- **Logs** provide us with the repository of data which are used to **detect patterns, anomalous behavior, and other signals indicative of click fraud.**
- filters (stage 1), which operate in real-time
- stages 2 and 3 which on deeper analysis of the data in our logs
- stage 2: pores over millions of impressions and clicks over longer time period
- **looking for unusual behavior** in hundreds of different data points
- **IP addresses:** for a given publisher or advertiser, where are their clicks coming from? Are they all coming from one country or city? Is that normal for an ad of this type?
- Don't identify individuals: **look at these in aggregate and study patterns**
- **Abnormally high number of clicks on a single publisher from the same ISP:** does look suspicious and raises a flag for us to investigate

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
A	$1.00	1%	1 cent
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
		Click through rate	Expected revenue

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
A	$1.00	1%	1 cent

Complications: Budget

- **Two complications:**
  - **Budget**
  - **Click-through rate (CTR) of an ad is unknown**

- **Each advertiser has a limited budget**
  - **Search engine guarantees that the advertiser will not be charged more than their daily or monthly budget.**

Complications: CTR

- **CTR: Each ad has a different likelihood of being clicked**
  - **Advertiser 1** bids \$2, click probability = 0.1
  - **Advertiser 2** bids \$1, click probability = 0.5
  - **Click-through rate (CTR)** is measured **historically**
    - **Very hard problem: Exploration vs. exploitation**
      Exploit: Should we keep showing an ad for which we have good estimates of click-through rate
      or
      Explore: Shall we show a brand new ad to get a better sense of its click-through rate.

Greedy Algorithm

- **Our setting: Simplified environment**
  - There is **1** ad shown for each query
  - All advertisers have the same budget ***B***
  - All ads are equally likely to be clicked
  - Value of each ad is the same (=**1**)

- **Simplest algorithm is greedy:**
  - For a query pick any advertiser who has bid **1** for that query
  - **Competitive ratio of greedy is 1/2.**

Bad Scenario for Greedy

- **Two advertisers A and B**
  - ***A*** bids on query ***x***, ***B*** bids on ***x*** and ***y***
  - Both have budgets of **$4**
- **Query stream: *x x x x y y y y***
  - Worst case greedy choice: ***B B B B*** **_ _ _ _**
  - Optimal: **A A A A B B B B**
  - **Competitive ratio = ½**
- **This is the worst case!**
  - **Note:** Greedy algorithm is deterministic – it always resolves draws in the same way.

Greedy algorithm with non-equal bids

- Greedy algorithm would **assign the query to**

a. **the highest bidder**
b. **who still has budget left.**

Greedy Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Greedy algorithm assigns the query to
a. the highest bidder
b. who still has budget

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q			
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

Greedy Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

Greedy Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

Greedy Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q	A2	0	40
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

Greedy Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q	A2	0	40
4th query q	A2	0	30
5th query q	A2	0	20
6th query q	A2	0	10
7th query q	A2	0	0
8th query q	No ad	0	0

BALANCE Algorithm [MSVV]

- **BALANCE** Algorithm by Mehta, Saberi, Vazirani, and Vazirani
  - **For each query, pick the advertiser with the largest unspent budget**
    - Break ties arbitrarily (**but in a deterministic way**).

Example: BALANCE

- **Two advertisers A and B**
  - **A** bids on query ***x***, **B** bids on ***x*** and ***y***
  - Both have budgets of **$4**

- **Query stream: *x x x x y y y y***

- **BALANCE choice: A B A B B B _ _**
  - Optimal: **A A A A B B B B**

- **In general:** For **BALANCE** on **2** advertisers
  Competitive ratio = ¾

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q			
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q			
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q	A2	0	10
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

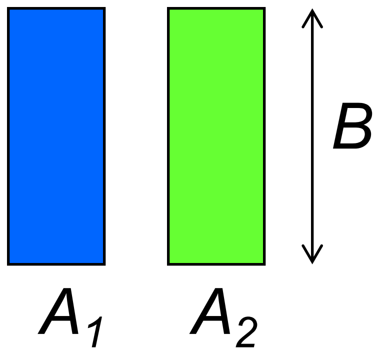
- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q	A2	0	10
7th query q	A2	0	0
8th query q	No Ad	0	0

Analyzing BALANCE

- **Consider simple case (w.l.o.g.):**
  - **2** advertisers, $\mathbf{A_1}$ and $\mathbf{A_2}$, each with budget **B** (≥1)
  - Optimal solution exhausts both advertisers' budgets
- **BALANCE must exhaust at least one advertiser's budget:**
  - **If not, we can allocate more queries**
    - Whenever both advertisers bid on the query, chosen advertiser's unspent budget only decreases
    - In BALANCE, one budget will be exhausted
- Assume BALANCE exhausts A_2's budget, but allocates x queries fewer than the optimal
- **Revenue:** $BAL = 2B - x$

Analyzing Balance



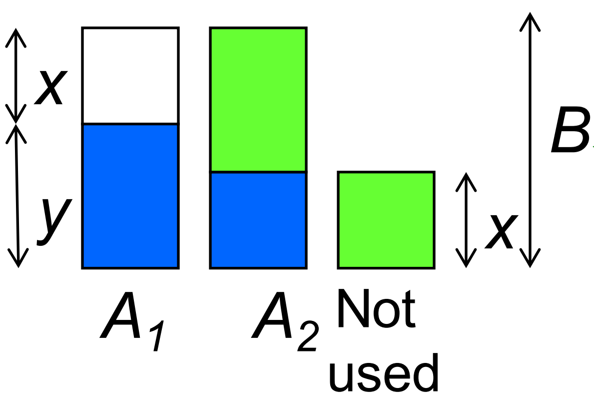
Queries allocated to A_1 in the optimal solution

Queries allocated to A_2 in the optimal solution

Optimal revenue = **2B**

Balance Algorithm:

Assume Balance gives revenue **= 2B-x = B+y**



Unassigned queries should be assigned to A_2

(if we could assign to $\mathbf{A_1}$ we would, since we still have budget)

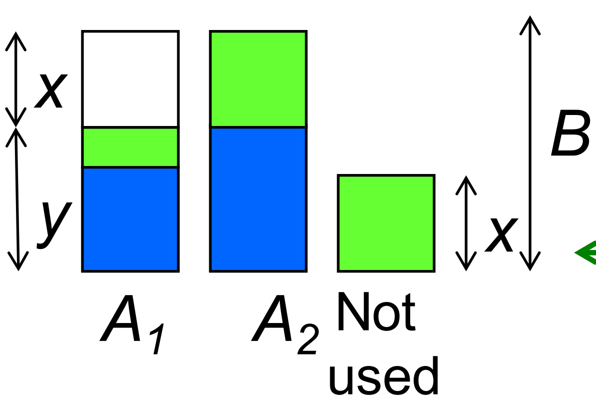
Goal: Show we have $\mathbf{y \geq x}$

Case 1) ≤ ½ of $\mathbf{A_1}$'s queries got assigned to $\mathbf{A_2}$

then y >= B/2, so surely y >= x

Case 2) > ½ of $\mathbf{A_1}$'s queries got assigned to $\mathbf{A_2}$

then x < B/2 and x + y = B so y >= x



Balance revenue is minimum for x=y=B/2

Minimum Balance revenue = 3B/2

Competitive Ratio = 3/4

BALANCE exhausts A_2's budget

BALANCE: General Result

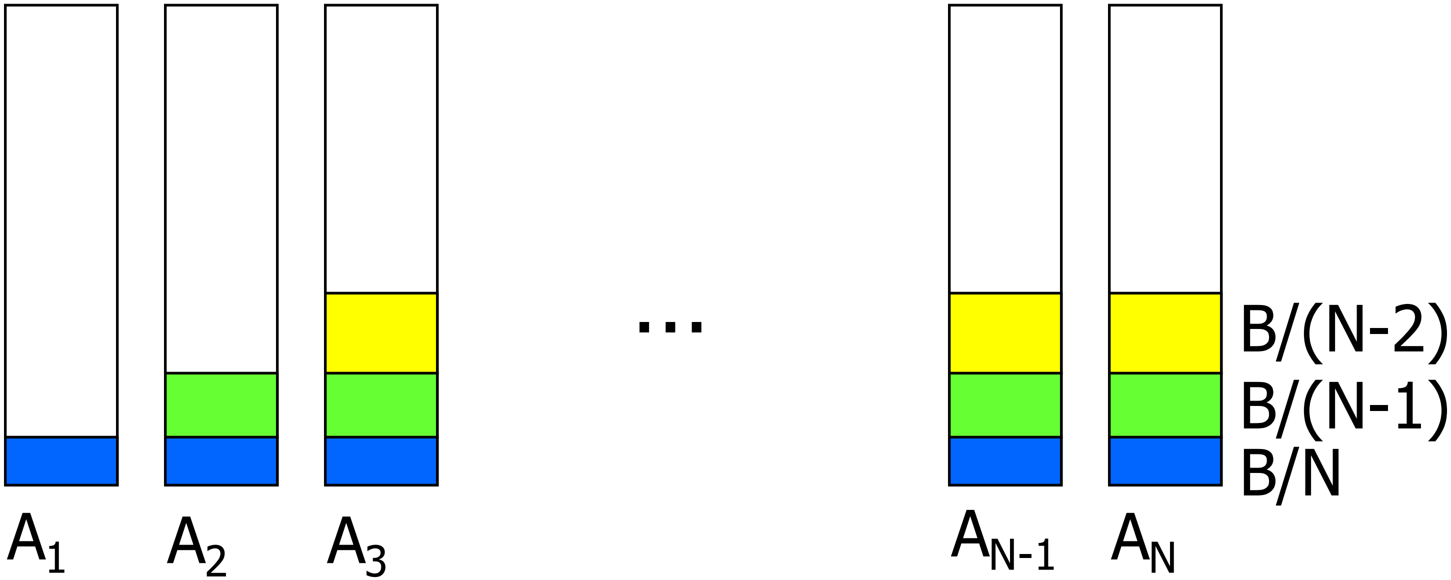
- **For Balance algorithm with many bidders**
- **In the general case, worst competitive ratio of BALANCE is 1–1/e = approx. 0.63**
  - Interestingly, no online algorithm has a better competitive ratio!

- **Let's see the worst case example that gives this ratio.**

Worst case for BALANCE

- ***N* advertisers:** $A_1, A_2, \ldots A_N$
  - Each with budget $B > N$
- **Queries:**
  - $N \cdot B$ queries appear in N rounds of B queries each
- **Bidding:**
  - Round **1** queries: bidders $A_1, A_2, \ldots, A_N$
  - Round **2** queries: bidders $A_2, A_3, \ldots, A_N$
  - Round *i* queries: bidders $A_i, \ldots, A_N$
- **Optimum allocation:**
  Allocate round *i* queries to A_i
  - Optimum revenue $N \cdot B$
- **BALANCE:**
  - Assigns **each query in round 1** to **N advertisers equally**, since all bid on q1
  - Prefers bidder with **largest remaining budget**
  - For **q2**, divided **equally among $A_2, A_3, \ldots, A_N$**
  - For each **query qi**, advertisers **$A_i, \ldots, A_N$ get queries**.

BALANCE Allocation



BALANCE assigns each of the queries in round 1 to **N** advertisers.

After ***k*** rounds, sum of allocations to each of advertisers $A_k, \ldots, A_N$ is

$$\boldsymbol{S_k = S_{k+1} = \cdots = S_N = \sum_{i=1}^{k-1} \frac{B}{N-(i-1)}}$$

If we find the smallest *k* such that $S_k \geq B$, then after *k* rounds we cannot allocate any queries to any advertiser

BALANCE: Analysis

B/1   B/2   B/3   …   B/(N-(k-1)) … B/(N-1)   B/N

S_1

S_2

S_k **= B**

1/1   1/2   1/3   …   1/(N-(k-1)) … 1/(N-1)   1/N

S_1

S_2

S_k **= 1**

BALANCE: Analysis

- **Fact:** $\boldsymbol{H_n = \sum_{i=1}^{n} 1/i \approx \ln(n)}$ for large $\boldsymbol{n}$
  - Result due to Euler

1/1 1/2 1/3 … 1/(N-(k-1)) … 1/(N-1) 1/N

ln(N)

ln(N)-1 **S_k = 1**

- $\boldsymbol{S_k = 1}$ implies: $\boldsymbol{H_{N-k} = ln(N) - 1 = ln(\frac{N}{e})}$
- We also know: $\boldsymbol{H_{N-k} = ln(N-k)}$
- **So:** $\boldsymbol{N - k = \frac{N}{e}}$
- **Then:** $\boldsymbol{k = N(1 - \frac{1}{e})}$

N terms sum to ln(*N*).
Last *k* terms sum to 1.
First *N-k* terms sum
to ln(*N-k*) but also to ln(*N*)-1

BALANCE: Analysis

- So after the first **$k=N(1-1/e)$** rounds, we cannot allocate a query to any advertiser

- **Revenue = $B \cdot N\ (1-1/e)$**

- **Competitive ratio = $1-1/e$**

General Version of the Problem

- **Arbitrary bids and arbitrary budgets!**
- Consider we have 1 query $\boldsymbol{q}$, advertiser $\boldsymbol{i}$
  - Bid = $\boldsymbol{x_i}$
  - Budget = $\boldsymbol{b_i}$
- **In a general setting BALANCE can be terrible**
  - Consider two advertisers $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$
  - $\boldsymbol{A_1}$: $\boldsymbol{x_1} = \mathbf{1}$, $\boldsymbol{b_1} = \mathbf{110}$
  - $\boldsymbol{A_2}$: $\boldsymbol{x_2} = \mathbf{10}$, $\boldsymbol{b_2} = \mathbf{100}$
  - Consider we see **10** instances of **q**
  - BALANCE always selects $\boldsymbol{A_1}$ and earns **10**
  - Optimal earns **100**

Modifications Needed to BALANCE Algorithm

- **Bias choice of ad in favor of higher bids**
- **Consider the fraction of budget remaining,** so we **bias toward using some of each advertiser's budget**
- More "risk averse": don't leave too much of any advertiser's budget unused.

Generalized BALANCE Algorithm

- **Arbitrary bids:** consider query q, bidder i
  - Bid = x_i
  - Budget = b_i
  - Amount spent so far = m_i
  - **Fraction of budget left over** $f_i = 1 - (m_i/b_i)$
- Define $\psi_i(q) = x_i(1-e^{-f_i})$ ψ *(psi)*
  - *bid * (1-e^{-(fraction of budget left)})*
- Allocate query q to bidder i with largest value of $\psi_i(q)$
- **Same competitive ratio (1-1/e) .**

Example 8.10

- **Bidder A_1: $x_1 = 1$, $b_1 = 110$**
- **Bidder A_2: $x_2 = 10$, $b_2 = 100$**
- First occurrence of query q: **fraction 1 of budgets b_1 and b_2 remain**
- $\psi_1(q) = x_1(1-e^{-f_1}) = 1(1-e^{-1}) = 1 - 1/e = 0.63$
- $\psi_2(q) = x_2(1-e^{-f_2}) = 10(1-e^{-1}) = 6.3$
- **So first q is awarded to A_2**
- **$\psi_2(q)$ decreases, but for the next 9 instances of q: $\psi_2(q) > \psi_1(q)$ and queries are awarded to A_2**
- For 10^{th} instance of q, **remaining fraction of budget b_2 is 1/10**
- $\psi_2(q) = x_2(1-e^{-f_2}) = 10(1-e^{-1/10}) = 0.95$, *which is* > 0.63
- **After 10 queries q, have spent all of A_2's budget, and additional queries q will be awarded to A_1**
- **Total revenue for 10 queries q = 100**
- **Generalized Balance Algorithm: Successfully biased toward higher bids, took into account fraction of budget remaining.**

Additional Observations

- Algorithm as described **does not account for possibility that click-through rate differs for different ads**
- **Multiply bid by CTR when computing ψ**

- Also can **consider historical frequency of queries**
  - **Use historical frequency to predict future frequency.**

Adwords Aspects Not in Our Model

Matching bids and search queries:

- **In our simplified model, advertisers bid on sets of words**
- An advertiser's bid is eligible to be shown for search queries with **exactly the same set of words as advertiser's bid**

- In reality, **Google, Yahoo, Microsoft all offer advertisers "broad matching": inexact matches of the bid keywords**
- **Examples: subsets, supersets, words with very similar meanings**

- Charge advertisers based on **complicated formulas that take into account how closely related the search query is to the advertiser's bids**
- **Proprietary algorithms.**

Adwords Aspects Not in Our Model

Charging Advertisers for Clicks

- **In our simplified model, when a user clicks on an ad, the advertiser is charged the amount they bid**
- Known as a **first-price auction**
- **In reality, search engines use a more complicated system known as a second-price auction**
- **Each advertiser pays approximately the bid of the advertiser who placed immediately behind them in the auction**
- Example: **First-place advertiser would pay the bid of the second-place advertiser plus one cent**
- **Less susceptible to being gamed by advertisers than first-price auctions**
- **Lead to higher revenues for search engines.**

Summary

- Greedy Algorithem
  - Greedy algorithm would **assign the query to the highest bidder who still has budget left**
    - **Competitive ratio of greedy is ½**
- BALANCE Algorithm [MSVV]
  - **For each query, pick the advertiser with the largest unspent budget**
  - **Competitive ratio of Balance is ¾**
    - **In the general case, worst competitive ratio of BALANCE is 1–1/e = approx. 0.63**
    - **budgets of higher-numbered advertisers exhausted**
    - **In a general setting, BALANCE can perform poorly**
- Generalized BALANCE Algorithm
  - **Bias choice of ad in favor of higher bids and Consider the fraction of budget remaining $\psi_i(q)$.**