BIG DATA ANALYTICS

Advertising on the Web

Outline

On-line advertising

- Web applications support themselves through advertising
- The most lucrative venue for on-line advertising: **search**
- The **"adwords" model** of matching search queries to advertisements
- The algorithms: greedy and "on-line"

Online Advertising

- Banner ads: Initial form of web advertising
- On-line stores: to maximize the probability that the customer will be interested in the product
- 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
  - They pay only if the ad is clicked on

Display Ads

- $\sim$ advertising in traditional media
- The fee is typically a fraction of a cent per impression

Specialized Display Ads

- Traditional media: newspapers or magazines for special interests
- $\Longrightarrow$ many specialized, low-circulation magazines
- An ad for golf clubs on sports.yahoo.com/golf has much more value per impression

Content Personalization

- Use information about the user
- E.g. Sally likes golf $\Longrightarrow$ show her an ad for golf clubs
  - She may belong to a golf-related group on Facebook
  - She may mention "golf" frequently in emails posted on her gmail account
  - She may spend a lot of time on the Yahoo! golf page
  - She may issue search queries with golf-related terms frequently
  - She may bookmark the Web sites of one or more golf courses.

Privacy issues

- People like the free services that are usually advertising-supported
- These services depend on advertising being much more effective than conventional ads
- Better to see things you might actually use
- Potential for misuse if the information

Search Advertising

- Introduced by Overture around 2000
  - Advertisers bid on keywords
  - When someone searched fors that keyword, the highest bidders' ads are shown
  - Advertiser is charged only if the ad is clicked on
- Similar system adopted by Google around 2002: **Adwords**

Challenges for the Search Advertising (1)

- Ads are displayed in response to query terms:
  - inverted index of words
  - the advertiser specifies parameters of the ad
- How to rank ads?
- "Most-recent first":
  - advertisers post small variations of their ads at frequent intervals
  - $\Longrightarrow$ discover ads that are too similar

Challenges for the Search Advertising (2)

- Measure the attractiveness of an ad:
  - Attractive ads will be clicked on more frequently
  - The position of the ad has great influence is clicked
  - Attractiveness may depends on the query terms
  - All ads deserve the opportunity to be shown until their click probability can be approximated closely

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?

Outline

The Adwords system

- Only a limited number of ads with each query
- Users: a budget for all clicks on their ads in a month
- Ordering ads: **by the amount they expected to receive for display of each ad**
  - The click-through rate based on the history of displays
  - The value of an ad = the bid $\times$ the click-through rate.

Adwords Problem

- **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.
  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$\times$CTR)
- **We need an online algorithm!**

Online algorithms

- **Classic model of algorithms:**
  - You get to see the entire input, then compute some function of it
  - In this context, "offline algorithm"
- **Online Algorithms:**
  - You get to see the input one piece at a time, and need to make irrevocable decisions along the way
  - Optimizing the output

The Adwords Innovation

Advertiser	Bid	CTR	Bid×CTR
A	\$1	1%	1 cent
B	\$0.75	2%	1.5 cent
C	\$0.5	2.5%	1.125 cent

Complications: Budget

- **Two complications:**
  - Budget
  - 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 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
- Clickthrough 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

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: $xxxxyyyy$
  - Worst case greedy choice: $BBBB\_\_\_\_$
  - Optimal: $AAAABBBB$
  - Competitive ratio $= 1/2$
- Note: Greedy algorithm is deterministic – it always resolves draws in the same way

BALANCE algorithm

- 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: $xxxxyyyy$
- BALANCE choice:$ABABBB\_\_$
  - Optimal: A A A A B B B B
  - Greedy: $BBBB\_\_\_\_$
- Competitive ratio $= 3/4$

General Version of the Problem

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

Generalized BALANCED

- 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 $\psi_i(q) = x_i(1 - e^{-f_i})$
- Allocate query q to bidder i with largest value of $\psi_i(q)$
- Competitive ratio $(1 - 1/e)$

Generalized BALANCED

- The click-through rate differs for different ads
- $\Longrightarrow$ multiply the bid by the click-through rate when computing the $\psi_i(q)$'s
- $\Longrightarrow$ maximize the expected revenue

Matching Bids and Search Queries

- Simplified model: advertisers bid on sets of words
- "Broad matching":
  - ad is eligible also for search queries that are inexact matches of the bid keywords
  - E.g., subset of keywords/ queries with very similar meanings
- Need to take into account how closely related the search query is to the advertiser's bid

Charging Advertisers for Clicks

- Simplified model:
  - A first-price auction
- **A second-price auction**:
  - advertiser pays approximately the bid of the advertiser placed immediately behind them in the auction
  - less susceptible to being gamed by advertisers than first-price auctions
  - higher revenues for the search engine

References

- J. Leskovec, A. Rajaraman and J. D. Ullman *Mining of Massive Datasets* (2014), Chapter 8