How to maximize the revenue of market

Defu Zhang
Department of Computer Science, Xiamen University
Department of Management Science, City University of Hong Kong
Email: dfzhang@xmu.edu.cn
August, 2013, Singapore
Contents

- What is market?
- Adwords market
- Generalizes online bipartite matching
- A practical project
  - Model
  - Algorithms
  - Simulation
  - Conclusion
- References
What is market?

Farm product market
Stock Markets
Internet
Revolution in definition of markets

New markets defined by

- Google
- Amazon
- Yahoo!
- Ebay
- Facebook
- Baidu
- Tencent
- Sina

......
Adwords Market

- Created by search engine companies
  - Google
  - Yahoo!
  - MSN

- Multi-billion dollar market

- Totally revolutionized advertising, especially by small companies.
The Adwords Problem

N advertisers;
- Daily Budgets $B_1, B_2, \ldots, B_N$
- Each advertiser provides bids for keywords he is interested in.
The Adwords Problem

N advertisers;
- Daily Budgets $B_1, B_2, \ldots, B_N$
- Each advertiser provides bids for keywords he is interested in.

queries (online) --> Search Engine
The Adwords Problem

N advertisers;
- Daily Budgets $B_1, B_2, \ldots, B_N$
- Each advertiser provides bids for keywords he is interested in.

queries (online)  
Search Engine  
Select one Advertiser pays his bid
The Adwords Problem

N advertisers;
- Daily Budgets $B_1, B_2, ..., B_N$
- Each advertiser provides bids for keywords he is interested in.

queries (online) \[\rightarrow\] Search Engine \[\rightarrow\] Select one Ad
Advertiser pays his bid

Maximize total revenue

Online competitive analysis - compare with best offline allocation
The Adwords Problem

N advertisers;
- Daily Budgets $B_1, B_2, ..., B_N$
- Each advertiser provides bids for keywords he is interested in.

queries (online) \[\rightarrow\] Search Engine \[\rightarrow\] Select one Ad
Advertiser pays his bid

Maximize total revenue

Example - Assign to highest bidder: only $\frac{1}{2}$ the offline revenue
Markets are so huge, big data!!!

- Massive computational power available for running these markets in a centralized or distributed manner

- Important to find good models and algorithms for these markets
Example

<table>
<thead>
<tr>
<th></th>
<th>Bidder 1</th>
<th>Bidder 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>$1</td>
<td>$0.99</td>
</tr>
<tr>
<td>CD</td>
<td>$1</td>
<td>$0</td>
</tr>
</tbody>
</table>

Queries: **100 Books** then **100 CDs**

\[ B_1 = B_2 = $100 \]

**Greedy Algorithm**

Lose

Revenue **$100**

Bidder 1       Bidder 2
Department of Computer Science, Xiamen university
### Example

<table>
<thead>
<tr>
<th></th>
<th>Bidder 1</th>
<th>Bidder 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>$1</td>
<td>$0.99</td>
</tr>
<tr>
<td>CD</td>
<td>$1</td>
<td>$0</td>
</tr>
</tbody>
</table>

Queries: 100 Books then 100 CDs

\[ B_1 = B_2 = $100 \]

Optimal Allocation

Revenue 199$

Department of Computer Science, Xiamen university
Generalizes online bipartite matching

- Each daily budget is $1, and each bid is $0/1.
Online bipartite matching

advertisers

queries

Department of Computer Science, Xiamen university
Online bipartite matching

advertisers     queries

Diagram showing the bipartite matching problem with advertisers on one side and queries on the other, connected by edges.
Online bipartite matching

Advertisers

Queries

Department of Computer Science, Xiamen university
Online bipartite matching

advertisers queries
Online bipartite matching

advertisers queries
Online bipartite matching

advertisers

queries
Online bipartite matching

Advertisers

Queries
Karp, Vazirani & Vazirani, 1990: 
\[ 1 - \frac{1}{e} \] factor randomized algorithm. Optimal!

Kalyanasundaram & Pruhs, 1996: 
\[ 1 - \frac{1}{e} \] factor algorithm for \( b \)-matching:
Daily budgets \( \$b \), bids \( \$0/1 \), \( b > > 1 \)
Algorithms for Adwords Problem


- Mehta, Saberi, Vazirani & Vazirani, 2005: $1 - \frac{1}{e}$ algorithm, assuming budgets $>>$ bids.
New Algorithmic Technique

- Idea: Use both bid and fraction of left-over budget

Give query to bidder with max bid $\times \psi$ (fraction of budget spent)

$$
\psi(x) = 1 - e^{-(1-x)}
$$
One practical project

- Users randomly initiate some events
- Some event meeting some constraints will be matched some campaigns. Every campaign will recommend some products. Product recommendation will produce some profits while recommendation is clicked.
It will be without constraints if recommendation is produced by web. The contacted times of each customer is restricted when recommendation via message, call center.

Some campaigns may have resource’s or budget’s restriction.

Some offers may have resource’s restriction.
Assume that the revenue of offer $i$ is $V(i)$, then

$$V(i)=C(i) \cdot P(i) \cdot \psi(T(i))$$

Where $C(i)$ denotes the revenue is obtained by offer $i$ clicked(or booked), $P(i)$ denotes the acceptance probability of offer $i$ that is computed by (clicking times/acceptance times). $T(i)$ denotes by (the execution times of treatment/the acceptance times of treatment).

$\psi(x) = 1 - e^{x-1}$ is as before.
Figure 1. Web recommendation model
Algorithm

Figure 2. Flowchart for Web recommendation
To select campaign and offer, the presented algorithm (Named maxCmaxB) is as follows:

- **First select** offer with the highest price $C_{\text{max}}$, then select the corresponding campaign with the highest left-over budget.
Simulation

- The kinds of events=30, Campaigns=100, offers=200, the total budget=173208 and the times of budget’s Offer*unit price=149459, the revenue for 20000 events is shown in Table 1.
## Table 1: The revenue of algorithms for more campaigns’ budget and more offer’s times

<table>
<thead>
<tr>
<th></th>
<th>2000</th>
<th>4000</th>
<th>6000</th>
<th>8000</th>
<th>10000</th>
<th>12000</th>
<th>14000</th>
<th>16000</th>
<th>18000</th>
<th>20000</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxCmaxB</td>
<td>15012</td>
<td>29078</td>
<td>42343</td>
<td>54071</td>
<td>64363</td>
<td>73346</td>
<td>81096</td>
<td>86859</td>
<td>91773</td>
<td>95617</td>
</tr>
<tr>
<td>maxCgoogleB</td>
<td>14962</td>
<td>28749</td>
<td>41730</td>
<td>52558</td>
<td>61741</td>
<td>70259</td>
<td>78063</td>
<td>83409</td>
<td>88674</td>
<td>93097</td>
</tr>
<tr>
<td>googleCmaxB</td>
<td>14645</td>
<td>27981</td>
<td>40386</td>
<td>51637</td>
<td>61279</td>
<td>69410</td>
<td>76863</td>
<td>82619</td>
<td>87356</td>
<td>91572</td>
</tr>
<tr>
<td>googleCgoogleB</td>
<td>14541</td>
<td>27662</td>
<td>39332</td>
<td>49864</td>
<td>58969</td>
<td>66586</td>
<td>73760</td>
<td>80343</td>
<td>85730</td>
<td>90490</td>
</tr>
</tbody>
</table>
Figure 3: The revenue of algorithms for more campaigns' budget and more offer's times.
Conclusions

- The proposed algorithm is more efficient.
- The model and algorithm should make a responding improvement when the application case (constraints change) changes.
references

references


Thanks for your attention!

Any questions?