Adaptive Weighing Designs for Keyword Value Computation

John W. Byers
Computer Science Dept.
Boston University & Adverplex Inc

Michael Mitzenmacher
School of Eng.& Appl. Sci.
Harvard University

Georgios Zervas
Computer Science Dept.
Boston University & Adverplex Inc
Keyword Value Per Click (VPC) Estimation

Keyword VPC = \frac{\text{Total keyword revenue}}{\text{Total keyword clicks}}

VPC is a foundational quantity in online advertising used to calibrate keyword bids.

Highly competitive environment, a small VPC error can make or break a campaign.
Channelization Setting

User performs query for “fax machines”
Channelization Setting

User clicks on paid ad

Advertiser charged per click

Keyword

Incoming click

$99.99

Purchase from Amazon.com

FaxMachines.com
Channelization Setting

User clicks on paid ad

Advertiser charged per click

Incoming click

Keyword

FaxMachines.com

$99.99

Purchase from Amazon.com
Channelization Setting

User clicks on paid ad

Advertiser charged per click

Incoming click

Keyword

FaxMachines.com

$99.99

Purchase from Amazon.com

Outgoing click

Channel

$99.99

Order fulfillment by 3rd party

Advertiser paid per conversion
Channelization Setting

User clicks on paid ad

Advertiser charged per click

Incoming click
Keyword

Order fulfillment by 3rd party

Advertiser paid per conversion

Outgoing click

Daily revenue reporting by channel

Incoming click

FaxMachines.com

Purchase from Amazon.com

$99.99

Outgoing click

Amazon.com

Buy Now!

$99.99
Channelization Setting

- User clicks on paid ad
- Advertiser charged per click
- Incoming click
- Keyword
- FaxMachines.com
- $99.99 Purchase from Amazon.com
- Outgoing click
- Channel
- Amazon.com
- $99.99
- Order fulfillment by 3rd party
- Advertiser paid per conversion
- Daily revenue reporting by channel
- Update keyword bids using VPCs
Channelization Setting

Abundance of keywords vs Scarcity of channels

Revenue from many keywords has to be bundled within the same channel
Channels

Each outgoing click is associated with a channel selected by the advertiser.

A channel is a place to aggregate keyword revenue.

Why is this important?
3rd-party revenue reporting is broken down by channel.
Problem Statement

- **Goal:** Obtain accurate VPC estimates per keyword

- We have daily channel measurements
  - **Known:** clicks per keyword, revenue per channel
  - **Unknown:** keyword VPCs
    - $|\text{keywords}| > |\text{channels}|$

- “Abusing” channels
  - keywords can be remapped to channels each day
Disaggregating aggregate data

- Regression methods are well-established
- Our contributions

**Unique setting:** High skew & scale, variances per keyword instead of per channel measurement

**Adaptivity:**
- We arrange measurements over time to speed up convergence
- Take cues from design of experiments [Hotelling 44]
  - Error associated with scale (equipment)
VPC Estimation

Straightforward if $|\text{keywords}| \leq |\text{channels}|$

Keyword 1
10 clicks

Channel 1
$50

VPC estimate
$5

Keyword 2
75 clicks

Channel 2
$75

$1

Keyword n
100 clicks

Channel n
$200

$2
What if $|keywords| >> |channels|$?

Idea 1: Round-Robin

Keyword 1 $\rightarrow$ Channel 1
Keyword 2 $\rightarrow$ Channel 2
. . .
Keyword k $\rightarrow$ Channel k

Keyword k+1 $\rightarrow$ Channel 1
Keyword k+2 $\rightarrow$ Channel 2
. . .
Keyword 2k $\rightarrow$ Channel k

Day 1 | Day 2 | ...  

Solution

$vpc_i = \frac{\text{Total revenue}}{\text{Total clicks}}$

Very inefficient!
Sparse clicks and conversions
What if $|\text{keywords}| > |\text{channels}|$?

Idea 2: Regular-$p$

Assign each keyword’s clicks and revenue to $p$ channels each day
(chosen uniformly at random, note that some channels might be left empty)

Now each keyword is measured daily
Regular-$p$ solution

• After enough measurements we are left with the following **overdetermined** system of equations

\[
\mathbf{r}_{jt} = \sum_{i \in c_{jt}} \text{clicks}_{ijt} \ast vpc_i
\]

where \(c_{jt} = \{\text{keywords in channel } j \text{ on day } t\}\)

• Solve using linear regression to get daily keyword VPC estimates: \(\hat{vpc}_{it}\)
Possible improvement: adaptivity (previous solutions oblivious)

Days 1 2 3 4 t-1 t

Estimate VPCs → Assign channels
Estimate VPCs → Assign channels
Estimate VPCs → Assign channels
Estimate VPCs → Assign channels
Estimate VPCs → Assign channels
Estimate VPCs

Use VPCs up to day $t$ to inform channel selection for day $t+1$
Adaptivity with Linear Regression

- **Algorithm:** Adaptive-OLS
- Try to distribute expected revenue uniformly among channels to avoid overfitting.
- Use least-full bin-packing to pack:
  
  \[
  E(\text{revenue}) = \hat{v}pc_i \times \text{avg}(\text{clicks}_i)
  \]
- We also tried packing VPCs, clicks, etc, but expected revenue performed best.
Methods so far...

<table>
<thead>
<tr>
<th></th>
<th>Weighted Averages (1kw/ch)</th>
<th>Linear Regression (many kw/ch)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oblivious</strong></td>
<td>Round-Robin</td>
<td>Regular-(p)</td>
</tr>
<tr>
<td><strong>Adaptive</strong></td>
<td>Adaptive-1</td>
<td>Adaptive-OLS</td>
</tr>
</tbody>
</table>
Experimental Setup

- VPC ($\mu_i$) and click ($\nu_i$) means follow a skewed distribution (Zipf 1.8)
- Variances proportional to means
- Normally distributed daily VPCs and clicks. On day $t$ keyword $i$ has:

$$vpc_{it} \sim N(\mu_i, \sigma_i^2), \text{ clicks}_{it} \sim N(\nu_i, \tau_i^2)$$
Comparing algorithms: Weighted RMSE

• Define the error on day $t$ as

$$e_t = \sqrt{\frac{1}{n} \sum_i v_i^2 (\mu_i - \hat{v}pc_{it})^2}$$

• Error is units of dollars ($\), suitable for optimizing bids
Adaptive vs Oblivious: Linear Regression
500 keywords and 10 channels

\[ \sigma^2 = 2\mu, \quad \tau^2 = \frac{1}{4}v^2 \]
Heteroskedasticity

- Least squares assumption: equal variances in measurement errors.

**Violated:** Errors per keyword, not channel

**Implication:** Inefficient estimates

- **Solution:** Feasible Generalized Least Squares

- Run OLS once
- Divide each equation by $\sqrt{\text{residual}}$
- Rerun OLS to get VPC estimates

Effectively discounts equations with high measurement error
### Algorithms overview

<table>
<thead>
<tr>
<th></th>
<th>Weighted Averages (1 kw/ch)</th>
<th>Linear Regression (many kw/ch)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oblivious</strong></td>
<td>Round-Robin</td>
<td>Regular-( p )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regular-( p )-FGLS</td>
</tr>
<tr>
<td><strong>Adaptive</strong></td>
<td>Adaptive-1</td>
<td>Adaptive-OLS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adaptive-FGLS</td>
</tr>
</tbody>
</table>
OLS vs FGLS
500 keywords and 10 channels

Low VPC and click variance:
\[ \sigma^2 = 2\mu, \quad \tau^2 = \frac{1}{4}v \]

High VPC and click variance:
\[ \sigma^2 = 4\mu, \quad \tau^2 = \frac{1}{2}v \]
Conclusion

• VPC computation is a foundational problem in online advertising

• We have demonstrated how to solve this problem in a setting with a limited number of channels, high noise and dimensionality

Applying similar methods in a real world setting with

• 100,000’s of keywords
• 100’s of channels

**Yielded a doubling of ROI**