



# Campus Talk: Targeting Ads

Sandeepan Banerjee, Google

# AdWords

- Multiple ads on keywords, down right-hand-side
- Charges based on position
- Online sales

The screenshot shows a Google search for 'rental' in Mozilla Firefox. The search results are displayed in two columns. The left column contains organic search results, and the right column contains sponsored links. Four red price tags are overlaid on the sponsored links, indicating the cost per click for each ad.

Ad Title	Price
Rental \$15	
Apartment Rentals \$12	
Apartment Finder \$10	
Snowmass Condo Rentals \$8	

# Definitions Soup

---

- **Page inventory** available page slots for ads
- **Keywords** terms entered in a search, bought by adv.
- **Impression** showing ad to a user
- **Creative** – the text/image/video that is shown
- **CPM** Cost Per Mille (1,000 impressions)
- **CPC** Cost Per Click
- **CTR** Click Through Rate (= Clicks/Impressions)

# Creating an AdWords Ad

The screenshot displays the Google AdWords interface. At the top left is the Google AdWords logo. On the right, there are links for 'Help' and 'Sign out', and a search bar for 'Jump to previous customer...'. Below the logo is a navigation bar with tabs for 'Ops Control Center', 'Campaign Management', 'Reports', 'Analytics', and 'My Account'. Under 'Campaign Management', there are sub-links for 'Campaign Summary', 'Tools', and 'Conversion Tracking'. A search bar for 'Search my campaigns:' is also present. The main content area shows the breadcrumb 'Campaign Summary > Seattle Condo > Ad Group #1' and '1 of 2 Ad Group(s)'. The ad group is titled 'Ad Group: Ad Group #1 5204121' and is in a 'Paused' state. A description for the ad is provided: 'Seattle View Penthouse', '2 Bed, 2 Bath in Fantastic Location Mountain Views, Huge private deck', and a URL 'www.badros.com/view-condo-fsbo.html'. There are also links for 'View History: this ad group' and 'Ad Group Approval Bin: Primary | Secondary | All'. Below this, there are tabs for 'Summary', 'Keywords', and 'Ad Variations'. The 'Keywords' tab is selected, showing a table of keywords with columns for 'Keyword', 'Status', 'Current Bid', 'Clicks', 'Impr.', 'CTR', 'Avg. CPC', 'Cost', and 'Avg. Pos'. The table lists 11 keywords, including 'seattle apartment', 'seattle condo', 'seattle condominium', 'seattle fremont apartment', 'seattle fremont real estate', 'seattle fremont condo', 'seattle luxury apartment', 'seattle fremont house', 'seattle luxury condo', and 'seattle luxury condominium'. A 'Total' row is also present at the top of the table.

Feb 14, 2003 to Jan 20, 2006 [Change range](#)

+ Add keywords: [Quick add](#) | [Keyword tool](#)    [Edit Keywords](#) | [Search this list](#)

[Edit Keyword Settings](#)   [Delete](#)    1 - 11 of 11 keywords.

<input type="checkbox"/> Keyword	Status [2]	Current Bid Max CPC	Clicks	Impr.	CTR	Avg. CPC	Cost	Avg. Pos
Total	Enabled	Default \$1.00 <a href="#">[edit]</a>	456	22,864	1.99%	\$0.46	\$209.47	3.0
<input type="checkbox"/> <a href="#">seattle apartment</a>	Active	\$1.00	125	5,634	2.21%	\$0.30	\$37.26	1.1
<input type="checkbox"/> <a href="#">seattle condo</a>	Active	\$1.00	143	2,906	4.92%	\$0.43	\$61.89	1.2
<input type="checkbox"/> <a href="#">seattle condominium</a>	Active	\$1.00	50	1,296	3.85%	\$0.33	\$16.54	1.3
<input type="checkbox"/> <a href="#">seattle fremont apartment</a>	Active	\$1.00	2	79	2.53%	\$0.06	\$0.12	1.1
<input type="checkbox"/> <a href="#">seattle fremont real estate</a>	Active	\$1.00	1	53	1.88%	\$0.81	\$0.81	4.0
<input type="checkbox"/> <a href="#">seattle fremont condo</a>	Active	\$1.00	2	47	4.25%	\$0.28	\$0.57	1.2
<input type="checkbox"/> <a href="#">seattle luxury apartment</a>	Active	\$1.00	2	24	8.33%	\$0.09	\$0.17	1.1
<input type="checkbox"/> <a href="#">seattle fremont house</a>	Active	\$1.00	1	22	4.54%	\$0.54	\$0.54	1.9
<input type="checkbox"/> <a href="#">seattle luxury condo</a>	Active	\$1.00	4	19	21.05%	\$0.55	\$2.19	1.5
<input type="checkbox"/> <a href="#">seattle luxury condominium</a>	Active	\$1.00	1	7	14.28%	\$0.18	\$0.18	2.0

# Specialized Search

---

- Given a query, find the best ads from over 100,000 advertisers
- How do you model utility to users?
  - Want high-quality, targeted ads, that generate revenue
  - Balance importance of high click-through-rate (CTR) with advertiser's willingness to pay
- Auction theory helps!

# Ranking Ads

---

Keyword: **skydive**

[Skydive with Us](#)

Only one accident last year.  
Have fun and play the odds!

[www.skydivewithus.com](http://www.skydivewithus.com)

CPC=\$0.40, CTR=2%

Effective CPM =  $\$0.40 \times 20 = \$8$

[Need Skydiving Insurance?](#)

We've got your back.  
Even if you lose, you win!

[www.skydiveinsurance.com](http://www.skydiveinsurance.com)

CPC=\$0.20, CTR=5%

Effective CPM =  $\$0.20 \times 50 = \$10$

# Ad Auction Ranking

---

- \$0.40 and \$0.20 are “bids” per click reflecting the *maximum* CPC the advertiser is willing to pay
- Insurance company could have bid \$0.16001 CPC  
had eCPM =  $\$0.16001 * 50 = \$8.005$   
and still gotten ranked #1
- So... we act as if they did:  
they pay only \$0.16/click, not \$0.20

# Let Advertisers Bid True Value

---

- The system acts in their best interest
- No need to increase their bid when someone else gets ranked ahead of them
- When there's no competition, you pay the minimum
- The minimum based on quality of the ad, based on a user-driven assessment



# Auction Basics

---

- English – “going going gone!”
- Dutch – price dropped until someone bites
- 1<sup>st</sup> price sealed – winner pays their bid

“Winners curse” • “Bid Shading”

→ Complicates selecting a bid



# Vickrey Auction

---

- 2<sup>nd</sup> price sealed – pay 2<sup>nd</sup> highest bid
- All 4 auctions have the same expected revenue for seller
- Vickrey has simplest bidding strategy:  
Just bid your true value  
(no bid shading, no winner's curse)
- Won Nobel Prize in Economics (1996)

# Engineering challenge: Predicting CTR

---

- Dizzying set of factors could affect clickthrough
  - Country, time of day, targeted text vs query, ...
- How does one automatically figure out which factor is more relevant?
  - How to update model quickly in face of change
  - How do you estimate CTR for not-yet-shown ads?

# The 10 billion-dollar JavaScript snippet...

---

```
<script type="text/javascript"><!--  
google_ad_width = 728; google_ad_height = 90;  
google_ad_format = "728x90_as"; google_ad_type = "text_image";  
//--></script>  
  
<script type="text/javascript"  
src="http://pagead2.googlesyndication.com/pagead/show\_ads.js">  
  
</script>
```

# The Power of Data applied to Contextual Targeting

---

- Conventional wisdom:
  - Given an order of magnitude increase in computational power...
  - ... you can solve previously impractical problems
- Unconventional wisdom
  - Given an order of magnitude increase in data...
  - ... you can solve previously unsolvable problems!
- Consider how to determine similarity between text:
  - How similar is “Kofi Annan” to “UN Secretary-General”?

# Traditional Information Retrieval Similarity

---

- Traditionally: Similarity is function of *term frequency within a document and across all documents*
- $TF(w)$  = frequency of term  $w$  in a document/query
  - Intuition: a word appearing more frequently in a text is more likely to be related to its “meaning”
- $IDF(w) = \log [N/n_w] + 1$ 
  - where  $N$  = # documents,  $n_w$  is # documents containing  $w$
  - Intuition: words appearing in many documents are generally not very informative (e.g., “the”)
- TFIDF: contribution of a term is product of quantities:  
 $TFIDF(w) = TF(w) \times IDF(w)$

# Using TFIDF to Measure Similarity

---

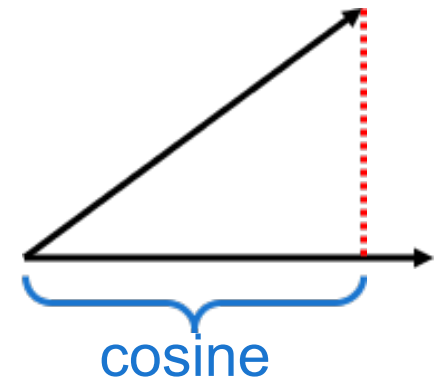
- Consider each document as a vector:

dog compute window ...

Doc. 1 =  $\langle 3.2, 0, 1.2, \dots \rangle$

Doc. 2 =  $\langle 0, 2.1, 5.4, \dots \rangle$

Doc. 3 =  $\langle 0, 1.7, 0, \dots \rangle$



- Vectors are constructed such that
  - Each dimension of vector represents a term  $w_i$
  - Each entry of vector has value:  $\text{TFIDF}(w_i)$
  - Normalize the vectors to unit length (Euclidean norm)
- Similarity of two texts is measured by the cosine between the TFIDF vectors of the documents/queries

○ Cosine = vector dot product!

# Determining Similarity of Short Text Snippets

---

- Many queries on the web are short (~2.5 words)
- For short text snippets, cosine is insufficient
- Cosine of term vectors for all following text pairs is 0:
  - “AI” “artificial intelligence”
  - “Kofi Annan” “UN Secretary-General”
  - “Eric Schmidt” “Google CEO”
  - “NASA” “space exploration”
  - “Larry Page” “Google founder”
- Should also identify unrelated concepts, even if high term overlap
  - “Larry Page” “web page”



# Determining Contextual Similarity of Short Text

---

*“... the meaning of a word is its use in the language”*

~ Ludwig Wittgenstein

- For short text snippets, need to determine greater contextual meaning
- Insight: leverage huge quantity of web information!
- Approach: Expand short text snippet into vector with additional context terms
  - Find terms that co-occur on web with terms in text snippet to determine contextual vector
  - Similar to “query expansion” in Information Retrieval

# Leverage the Web to Determine Similarity

---

- Let  $x$  and  $y$  be two short text snippets
- Want to define a function  $f(x, y)$  that measures “semantic” similarity between  $x$  and  $y$
- Define “query expansion” of text  $x$ ,  $\mathbf{QE}(x)$ , as follows:
  - Issue  $x$  as query to search engine (oh, say, Google...)
  - Let  $R$  be retrieved set of  $N$  documents:  $\{D_1, \dots, D_N\}$
  - Compute TFIDF vector  $\mathbf{V}_i$  for each document  $D_i \in R$
  - Compute  $\mathbf{QE}(x)$  as average (centroid) of all vectors  $\mathbf{V}_i$
- Define  $f(x, y) = \mathbf{QE}(x) \cdot \mathbf{QE}(y)$

# How Well Does This Work?

- Recall previous text pairs:  $f(x,y)$  Cosine

("AI", "artificial intelligence")	0.831	0.000
("Kofi Annan", "UN Secretary-General")	0.825	0.000
("Eric Schmidt", "Google CEO")	0.845	0.000
("NASA", "space exploration")	0.691	0.000
("Larry Page", "Google founder")	0.770	0.000
("Larry Page", "web page")	0.123	0.500
- Consider multi-faceted term "Java":

("Java island", "Indonesia")	0.454	0.000
("Java programming", "Indonesia")	0.020	0.000
("Java programming", "applet development")	0.563	0.000

# More Info

---

- Sandeepan Banerjee works on Google's Infrastructure (Crawling, Indexing, Storage, Map/Reduce, Video, International projects, ...)
- Send follow-up questions to [sandeepan@google.com](mailto:sandeepan@google.com)