

Web Usage Mining

Dr. Barry Shepherd
Institute of Systems Science
National University of Singapore
Email: barryshepherd@nus.edu.sg

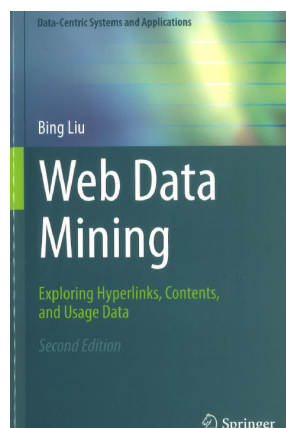


© 2018 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of ISS, NUS, other than for the purpose for which it has been supplied.

Web Usage Mining Definition

- The automatic discovery and analysis of patterns in clickstreams, user transactions and other associated data collected or generated as a result of the user's interactions with the Web
- Capture , model and analyze the behavioral patterns and profiles of users interacting with a website

A good reference



<https://cs.famaf.unc.edu.ar/~laura/lIibres/wm.pdf.gz>

Types of Data Used

- Web Server Logs
 - Page Click, Search Query, Search Engine Toolbars, Ad Impression, Ad Clicks,..
 - UserID is usually not known (rely on cookie, IP address, user agent etc.)
- Application Server Logs
 - e.g. ecommerce transactions
 - UserID is usually known - users are logged-in
- User Registration Data
 - Required fields + optional fields (basic demographics)
- User Volunteered Data
 - e.g. user ratings on products, downloads made, other volunteered info
- Third party data
 - e.g. demographics (at postcode, household and individual granularity)

Day Agenda

- Web Server Logs ~ structure and issues
- Web Server Log mining
- Search Query Log mining
- Behavioral targeting and Ad-click Prediction
- Assignment1

Web Server Logs

- Each record in the Web Server log file represents a single HTTP request
- A browser may fire multiple HTTP requests to Web server to display a single Web page

1. IIS (Internet Information Service) Samples: Here are some sample records from an IIS server log file:

```
02:49:12 127.0.0.1 GET / 200
02:49:35 127.0.0.1 GET /index.html 200
03:01:06 127.0.0.1 GET /images/sponsered.gif 304
03:52:36 127.0.0.1 GET /search.php 200
04:17:03 127.0.0.1 GET /admin/style.css 200
05:04:54 127.0.0.1 GET /favicon.ico 404
05:38:07 127.0.0.1 GET /js/ads.js 200
```

The record format is very simple. It has fields for: time, client IP address, request command, requested file, and response status code.

Sample Records from an Apache Server Log

- The record format is more complex. The records are also very long. Some fields require unpacking to obtain useful info

```
192.168.198.92 - - [22/Dec/2002:23:08:38 -0400] "GET
/images/logo.gif HTTP/1.1" 200 807 www.yahoo.com
"http://www.some.com/" "Mozilla/4.0 (compatible; MSIE 6...)" "-"
192.168.72.177 - - [22/Dec/2002:23:32:14 -0400] "GET
/news/sports.html HTTP/1.1" 200 3500 www.yahoo.com
"http://www.some.com/" "Mozilla/4.0 (compatible; MSIE ...)" "-"
192.168.72.177 - - [22/Dec/2002:23:32:14 -0400] "GET
/favicon.ico HTTP/1.1" 404 1997 www.yahoo.com
"-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.7.3)..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:15 -0400] "GET
/style.css HTTP/1.1" 200 4138 www.yahoo.com
"http://www.yahoo.com/index.html" "Mozilla/5.0 (Windows..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:16 -0400] "GET
/js/ads.js HTTP/1.1" 200 10229 www.yahoo.com
"http://www.search.com/index.html" "Mozilla/5.0 (Windows..." "-"
192.168.72.177 - - [22/Dec/2002:23:32:19 -0400] "GET
/search.php HTTP/1.1" 400 1997 www.yahoo.com
"-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; ...)" "-"
```

<http://www.herongyang.com/Windows/Web-Log-File-IIS-Apache-Sample.html>

Example: An E-Retailer Webserver log

Data from PKDD Cup 20005

```
10;1074585601;62.168.30.38;69c6e7a42f4a1652d26ab99ea69b64ba;/ls/?id=3;http://www.shop1.cz/ct/?c=141
14;1074585603;194.228.38.198;a4861ee652018db68444c1b0e69f0494;/ct/?c=505;http://www.shop4.cz/znacka/?c=39
14;1074585603;212.27.217.148;1c93382f635822e9de20cf85ae42099f;/ls/?id=118&view=1,2,3,9&pozice=40;http://www.shop4.cz/
12;1074585604;57.66.66.138;17bff4c98f96413dbe748c9cd8822da9;/ls/?id=38;http://www.shop3.cz/ct/?c=158
12;1074585602;212.158.128.176;d0a9a9e520d0670a77354c20c692df5e;/dt/?c=12953;http://www.shop3.cz/ls/?filtr=125&id=38&
11;1074585605;194.228.244.187;46b4037b211def7f54fcaa9ed9ece8fb;/poradna/?th=23421&ths=1;http://www.shop2.cz/poradna/
14;1074585605;213.151.84.126;4706abe8c12acb761d8a5e4eb29b3c2f;/ls/preber.php;http://www.shop4.cz/ls/?id=46
14;1074585605;195.250.143.86;e00397cd82cd28d6b32881d1c801640c;/ls/sety.php?id=164;http://www.shop4.cz/dt/?c=2514
14;1074585604;62.24.64.12;95ee0e53f3b1b390555fd069e560d65e;/dt/?c=10990;http://www.shop4.cz/ls/?filtr=42,41&id=74&so;
10;1074585605;158.196.177.79;cbf84093e4740423436abaf3c1a65ebc;/ct/?c=141;http://www.shop1.cz/
14;1074585605;194.228.38.198;a4861ee652018db68444c1b0e69f0494;/ls/?id=106;http://www.shop4.cz/ct/?c=505
10;1074585605;212.96.166.1;0031877ce5e977c08227de13ec65ce9e;/ls/index.php?id=4&view=1,8,10,11,29,36&filtr=9;http://
14;1074585605;213.151.84.126;4706abe8c12acb761d8a5e4eb29b3c2f;/ls/?filtr=34&id=46&sort=&view=1,2,3,6,4,16,5,12;http://
10;1074585606;194.108.220.148;fe830493a6804afbd8c1ed1acblaaace2;/akce/?kat=5;http://www.shop1.cz/akce/
16;1074585606;194.228.57.1;d8f96350d0c1756b65d26e684b95c8df;/ls/?id=139;http://www.shop6.cz/
14;1074585606;212.27.217.148;1c93382f635822e9de20cf85ae42099f;/dt/?c=8380;http://www.shop4.cz/ls/?id=118&view=1,2,3
14;1074585606;194.196.100.86;e9455a109435408eb7b8e170d636d024;/klient/stav.php?id=46104&srv_id=1;http://www.shop4.cz/
12;1074585607;194.196.251.162;3cd540088a904ca748b6cfd0e9ee2c2e;/ct/?c=155;http://www.shop3.cz/kosik/udaie.php
10;1074585607;158.196.177.79;cbf84093e4740423436abaf3c1a65ebc;/ls/?id=3;http://www.shop1.cz/ct/?c=141
14;1074585607;212.65.210.242;883fcf0c6e6a8b941d512df6ed3a40d1;/dt/?c=3485;http://www.shop4.cz/ls/index.php?id=62&vi;
17;1074585608;193.165.170.245;ed56a0a2c8bc7d74e0a4a65b556b75d1;/dt/?c=15379;http://www.shop7.cz/ls/?id=155&view=1&2(
11;1074585606;66.77.73.176;667141ac83c3916b7095107b155f3c8e;/dt/?c=9354;
14;1074585609;194.228.38.198;a4861ee652018db68444c1b0e69f0494;/ls/preber.php;http://www.shop4.cz/ls/?id=106
10;1074585609;62.168.30.38;69c6e7a42f4a1652d26ab99ea69b64ba;/ls/preber.php;http://www.shop1.cz/ls/?id=3
```

5 compound fields delimited by ‘;’

Web Logfile Data Issues

- Web Logs are messy, require much preprocessing & cleaning
 - High granularity – every event (HTTP request) logged
 - Duplications – same user activity can trigger multiple logged events
 - Complex, composite fields, e.g. long URL strings – parse to extract usable data
 - Bots – up to 30% of events can be triggered by bots, need removal
- Integration – Web Logs are commonly spread over multiple servers
 - Event ID's may differ- how to map diverse events to the same user/session?
 - Duplications & Conflicts – same activity can be logged in different log files
 - Time/Date synchronisation - different time zones, different date-time formats
- Excessive Detail & Large Size
 - Often excessive detail is logged – e.g. each image download in a page is logged
 - Removal of extraneous, irrelevant information, e.g. links to style files
- Sessionisation
 - identify all events belonging to the same user/site visit

Web Server Log Mining

- **Clickstream Mining:** What pages are frequently viewed together in the same session? What are the common page/event paths (clickstreams)?
- Site Optimisation
 - E.g. Ensure that pages viewed together are easy accessible from each other. Useful in e-retail to facilitate a smooth purchase path
- Understanding User Needs
 - Which users have similar browsing patterns? have similar interests/goals?
 - What are the browsing patterns of similar users?
E.g. compare guests with registered users – to encourage guests to register
- Make Recommendations
 - Help users achieve a goal, e.g. show fees page to a user viewing MTech details.
Typically use current session info only (short-term user profile)
 - Recommend other content, e.g. news, articles, products.
Can use longer-term user profile (many sessions) if available

Clickstream Mining – Approaches

- **Clickstream patterns and content recommendations**
 - **Association Mining** - find common associations between pages
e.g. home, news1, news2, sports12, fin4 (in any order) ► tech5
 - **Sequence Mining** – find common sequences of pages
e.g. home ► news1 ► news2 ► sports12 ► fin4 ► tech5
- **Content recommendations**
 - **Content-Based Approach** – find pages similar to those the user has read
 - **Collaborative Filtering** – find other users with similar browsing profiles to the current user, recommend what they like

Association Mining

- Treat web pages (typically in the same session) as items in a shopping basket - then use Market Basket Analysis (MBA) techniques
- A brute force approach is to count all possible item-sets
 - Items that occur together (e.g. are in the same basket) are called an item-set. If an item-set has a large count (a frequent item-set) then there is a potential association between the items in the item set

E.g. pages visited on a news site:

session1: home, news, sport
session2: finance, news
session3: fashion, home
session4: news, finance, home
session5: sport, home, finance
session6: fashion, home, news
session7: home, finance, news, sport

A frequent item-set is {home, news}

Co-occurrence counting for pairs :

	home	sport	news	fashion	finance
home	6	3	4	2	3
sport		3	2	0	2
news			5	1	3
fashion				2	0
finance					4

Symmetrical matrix - Items in each basket have no temporal ordering, we only need consider the green

Association Rule Learning

- Co-occurrence counting is not viable for large datasets and large item-sets
- Algorithms such as Apriori (Agrawal et al, '93) use heuristics to reduce the combinatorial size of the search space
 - If an item-set is frequent, then all of its subsets must also be frequent
 - The user specifies minimum rule support and rule confidence
- The found associations are expressed as rules, e.g.

session1: home, news, sport
session2: finance, news
session3: fashion, home
session4: news, finance, home
session5: sport, home, finance
session6: fashion, home, news
session7: home, finance, news, sport

Possible Rules

{home} → {news},
{home, news} → {finance},
{news} → {home, sport},

Itemset count

4
2
2

Note1: rules indicate co-occurrence, not causality or a sequence over time

Association Rule Metrics

- Rule Support
 - The proportion of transactions that contain the item set (LHS + RHS items)
- Rule Confidence
 - The proportion of rule firings that are correct predictions
 - Support for combination (LHS & RHS) / Support for condition (LHS)

$$= \frac{\text{\#transactions containing all items in the rule (LHS and RHS)}}{\text{\#transactions containing all items in the rule condition (LHS)}}$$

session1: home, news, sport
 session2: finance, news
 session3: fashion, home
 session4: news, finance, home
 session5: sport, home, finance
 session6: fashion, home, news
 session7: home, finance, news, sport

home → news
 Support = 4/7
 Confidence = 4/6

news → home
 Support = 4/7
 Confidence = 4/5

Note2: {news}→{home} may not have same confidence as {home}→{news}

e.g. whisky->coke (often), but coke->whisky(less)

Association Rule Metrics

$$\text{Rule lift} = \frac{\text{Support (LHS \& RHS)}}{\text{Support (LHS) * Support (RHS)}} = \text{the ratio of the observed support to that expected if LHS and RHS were independent}$$

If lift = 1 then this implies that the probability of occurrence of X and Y are independent (no association)
 If the lift is > 1, then LHS and RHS are dependent on each other (one makes the other more likely)
 If the lift is < 1, then one (LHS or RHS) has a negative effect on presence of other

session1: home, news, sport
 session2: finance, news
 session3: fashion, home
 session4: news, finance, home
 session5: sport, home, finance
 session6: fashion, home, news
 session7: home, finance, news, sport

home → news or news → home

$$\text{Lift} = \frac{4/7}{((6/7)*(5/7))} = 0.57 / 0.61 = 0.93$$

https://en.wikipedia.org/wiki/Association_rule_learning#Lift

Sequence Pattern Mining

- Similar to Association Mining but with sequence information added
 - E.g. A sequence of events over time, DNA sequences, word sequences
 - Many algorithms based on extended apriori, e.g. SPADE
- Mining Goal = Given a set of sequences, find all *frequent* subsequences
- Many more combinations to consider since $A \rightarrow B$ is not same as $B \rightarrow A$

In general:

sequences
<u>a</u> (<u>abc</u>)(<u>ac</u>)d(cf)
(ad)c(bc)(ae)
<u>a</u> (ef)(<u>ab</u>)(df) <u>cb</u>
eg(af)cbc

Items in brackets are unordered
(e.g. occur at the same time)

Frequent sub-sequence:

<a(ab)c>

Clickstreams (website visit sessions):

session1: home, news, sport, finance
 session2: finance, news
 session3: fashion, home, sport
 session4: news, finance, home
 session5: sport, home, finance
 session6: fashion, home, news, news, sport
 session7: home, finance, news, finance, sport

Frequent sub-sequence:

home-> news -> sport

Example: MSNBC Website Analysis

- Data from the Pagview log of MSNBC.com (a news site) on 28 Sep'99
 - Raw data ~ Datetime, URL, cookieID, IPaddress, UserAgent(browser) etc...

Preprocessing

- **Page view categorization** – the pages viewed (URLs) were first converted into topic categories
- **Grouping by unique user** - each data row shows the sequence of pageview categories for one user on that day

Codes for the msnbc.com page categories

category	code	category	code	category	code
frontpage	1	misc	7	summary	13
news	2	weather	8	bbs	14
tech	3	health	9	travel	15
local	4	living	10	msn-news	16
opinion	5	business	11	msn-sport	17
On-air	6	sports	12		

Sequences:

```
6
1 1
6
6 7 7 7 6 6 8 8 8 8
6 9 4 4 4 10 3 10 5 10 4 4 4
1 1 1 11 1 1 1
12 12
```

Number of users: 989,818
 Average number of visits per user: 5.7
 No. of URLs per category: 10 to 5000

Example MSNBC Rules*

- Association Rule Examples

on-air & business & sports & bbs	--> frontpage	86.22%
news & tech & misc & bbs	--> frontpage	86.18%
on-air & misc & business & sports	--> frontpage	86.16%
tech & misc & travel	--> on-air	86.09%
tech & living & business & sports	--> frontpage	86.08%
news & living & sports & bbs	--> frontpage	85.99%
misc & business & sports	--> frontpage	85.79%

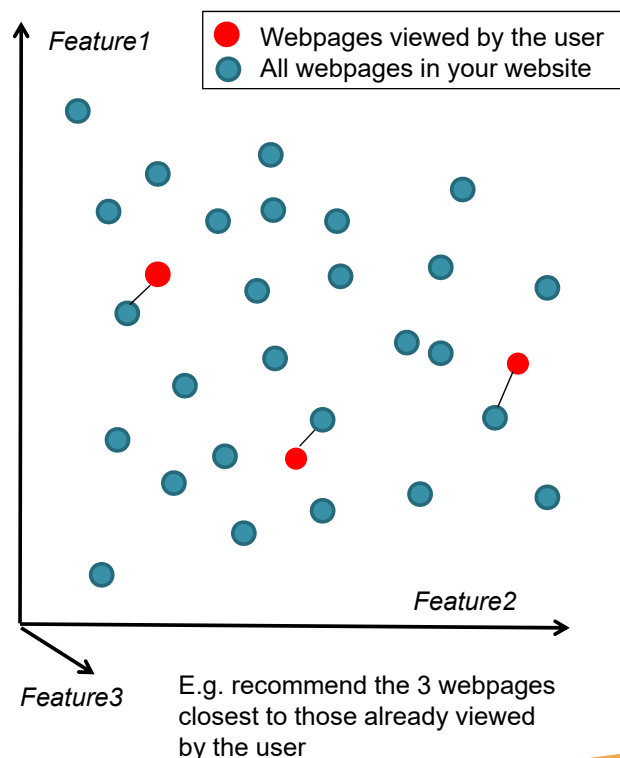
- Sequence Rules Examples

on-air → news	1.51%
news → frontpage → news	1.49%
local → news	1.46%
frontpage → frontpage → business	1.35%
news → sports	1.33%
news → bbs	1.23%
health → local	1.16%

(*Frequent Pattern Mining in Web Log Data, Ivancsy, Vajk, 2006.
Using their own association and sequence finding algorithms)

Content-Based Page Recommendation

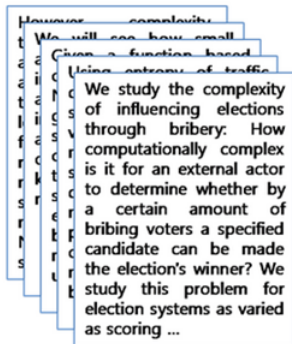
- Look at the content within the webpages
- Simple Approach
 - Categorise your webpages and ask the user directly what categories they prefer (e.g. news, music, fashion...)
- Vector-Space Approach
 - Model users and webpages as vectors
 - Look for similarities between the user and webpages
 - For **new users**: ask the user to rate a selection of pages / products
 - For **existing users**: look at the webpages the user has already viewed



Document Modelling

- Bag-of-Words Approach
 - Count the keywords and key phrases in each document and put into a vector
 - Use a distance metric (e.g. cosine distance) to measure distance between vectors

Documents



	complexity	algorithm	entropy	traffic	network
D1	2	3	1		
D2				2	1
D3	3			3	4
D4	2	4	2		
D5	3	4			

Document-term matrix (DTM)

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Cosine similarity ranges from -1 (exactly opposite) to 1 (exactly the same).
It's good because it ignores magnitude (e.g. doc size)



	word1	word2	word3	word4	word5
Doc1	0	10	20	100	200
Doc2	0	100	200	1000	2000

Cosine Similarity = 1
Euclidean Dist. = 2022.5

Document Modelling

- Use **N-Grams** not raw words: word sequences (typically N = 2 or 3)

“find me cheap traveler’s hotels in London” (raw query)
 “cheap travelers hotels in London” (tokenization)
 “cheap travelers hotels London” (stop word removal)
 “cheap travel hotel London” (stemming)
 “cheap travel”, “travel hotel”, “hotel London” (2-grams)



- Use **Term Frequency (TF)** not raw counts

- Normalise so that big documents don't dominate

$$= \frac{\text{frequency of the term in the document}}{\text{max frequency of any term in the document}}$$

- **Inverse Document frequency (IDF)**

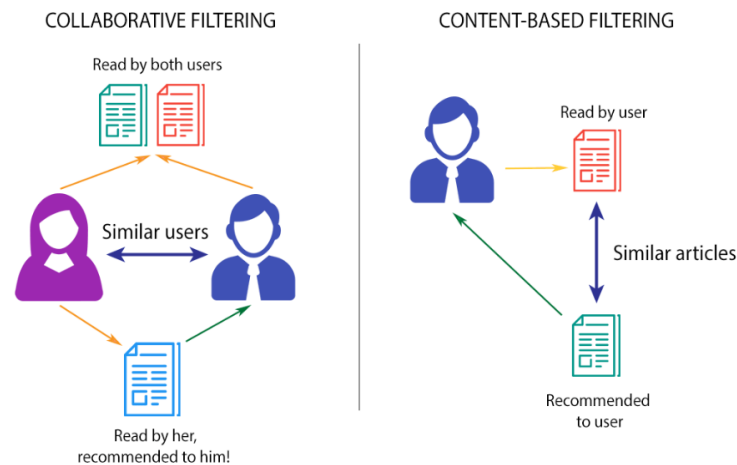
- if a term appears in many documents then its probably not important, or useful for discrimination

$$= \frac{\log(\# \text{Documents})}{\# \text{Documents containing the term}}$$

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

Collaborative Filtering (CF)

- Match users to people with similar tastes – recommend what they like
- Doesn't require analysis of content (product properties, page content etc)



Collaborative Filtering (CF)

- Collaborative filtering is often used with user-ratings data
 - E.g. Users rate products or content on a scale of 1 to 5
- Basic CF Algorithm (User-Based CF)

User	movie1	movie2	movie3	movie4	movie5	movie6	etc.
1	2	5	4		3	1	
2		3		5	3	1	
3			5	3			

*The
"Ratings"
matrix*

- Find the closest users to the target user (use Cosine similarity or Pearson coefficient*)
- Recommend the item that has the highest weighted average rating among these neighbours (weight by the distance to the target user)
- *Using Pearson Coefficient is equivalent to first subtracting each users mean rating from their individual ratings – this removes any bias due to some people always rating higher or lower

CF Using Page View Logs

- Rather than asking for explicit ratings of webpages we could assume
 - Repeat visits to a page implies liking (more repeats => more likes)
 - More time on page implies liking (longer duration => more like)

User (or Session)	page1	page2	page3	page4	page5	page6	etc.
1	2	5	4		3	1	
2		3		5	3	1	
3			5	3			

Assume the ratings here refer to the time spent on page (normalised to 1-5)

- Find the closest users to the target user
OR Find the closest session to the current session
- Recommend the page that has the highest weighted average rating among these neighbours (weight by distance to the target user or session)
- Can also apply linear-decay (or exponential) to decay the ratings by the time since the page was viewed – this introduces “sequence information” into the calculation

CF Using Page View Logs

- Item-Based CF
 - Find the closest page to the current page being viewed
 - Faster to execute since item properties do not change very quickly hence can precompute the item-item similarity matrix (e.g. precompute overnight)

Item (Page)	session1	session2	session3	session4	session5	session6	Etc..
1	2	5	4		3	1	
2		3		5	3	1	
3			5	3			



Cornell University
Library

Evaluation of Session-based Recommendation Algorithms

Malte Ludwig, Dietmar Jannach

(Submitted on 26 Mar 2018)

<https://arxiv.org/abs/1803.09587>

This also introduces
factorization and deep
learning approaches to
recommender systems

CF Using Page View Logs

- Clustering can also help reduce computational load
- E.g. cluster users based on their past page-views
- Compute the cluster centroids by averaging the weights (e.g. #views) for each page
 - E.g. if the page weights are boolean ~ was the page viewed in last month (T/F), then

		P1	P2	P3	P4	P5	P6
cluster1	User7 User53						
cluster2	User19 User8 User45 User11	1 1 1 0	1 1 1 1	0 0 0 1	0 0 0 0	0 0 0 0	1 1 1 1
cluster3	user22 user17						

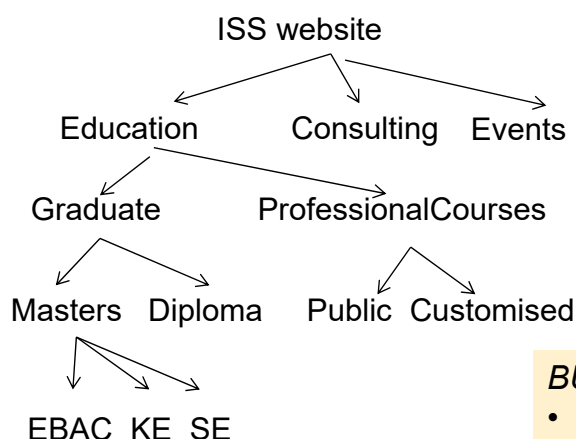
Cluster2 Profile	
Weight	Page
1	P2
1	P6
0.75	P1
0.25	P3

- To make a page recommendation we find the nearest centroid to the user and recommend the pages in that cluster that have high weights

See ch12, p469, Web Data Mining, Bing Liu

Clickstream Mining Issues

- Finding associations between very granular events (e.g. detailed URL's) is often unsuccessful due to the sparsity of the individual events. Finding associations between higher level concepts may be more successful. Similarities between clickstreams may also be easier to find at a less granular level. Travelling up taxonomies can help overcome this...



BUT new Issues now arise.....

- Which taxonomy to use? – many may apply
- What level(s) in the taxonomy to use?
- How to categorise the raw events?

Clickstream Mining Issues

- Many page-views that actually occurred are often missing in clickstream logs (use of back button, pages held in cache, pages held on other server)

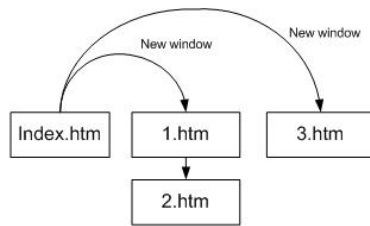


Figure 1: User opening multiple windows into the same site

Rec. No.	URL	Referrer
1	Index.htm	-
2	1.htm	Index.htm
3	2.htm	1.htm
4	3.htm	Index.htm

Table 2a: Clickstream before PRM reconstruction

Rec. No.	URL	Referrer
1	Index.htm	-
2	1.htm	Index.htm
3	Index.htm	-
4	2.htm	1.htm
5	Index.htm	-
6	3.htm	Index.htm

Table 2b: Clickstream after PRM reconstruction

Many methods:


E.g. Pattern Restore Method (Ting *et al.* 2005) attempts to reconstruct missing server-side clickstream data based on referring site information and the Website's link structure.

A record is assumed missing if either:

- referrer URL != target URL
- there is no direct link from previous page to current page in the website structure

See <http://www.informationr.net/ir/11-2/paper249.html>
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.7143>

Day Agenda

- Web Server Logs ~ structure and issues
- Web Server Log mining
- Search Query Log mining 
- Behavioral targeting and Ad-click Prediction
- Assignment1

Search Query Logs: Structure & Issues

Timestamp	Date-time
IP address of client	but not if proxy or ISP access
User Agent	type & version number of the browser
CookiedID	search engines leave a cookie on the client machine
The query string	The search query types in by the user
Result List	the URLs returned by the search engine
Clicked URL	which of the above URLs was clicked on

Ambiguity

- Most queries are short, typically a few words – hard to infer user intent
- Many queries have multiple meanings.
e.g., “apple”, “java”

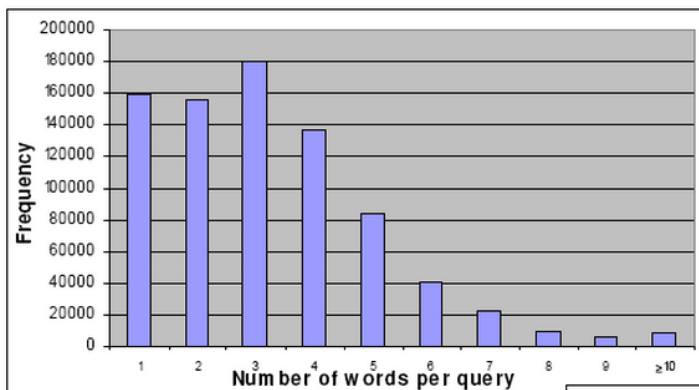
Frequently changing terms

- Many queries are product, fashion & news related which change frequently

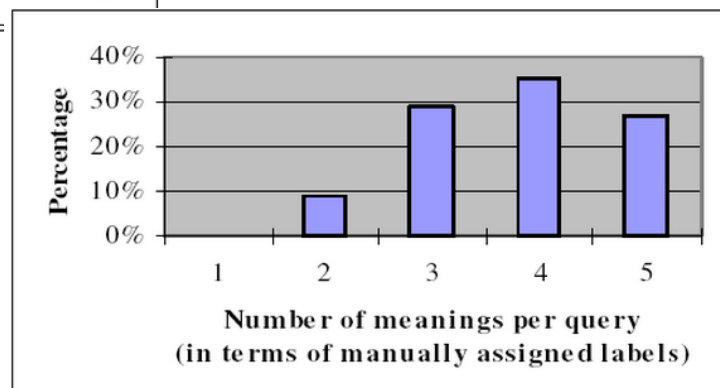
Dominance by Common Terms

- Queries have a very long tail distribution. A few words are very common, the rest very rare. Roughly follow the **Zipf** law

KDD'05 Cup: Search Query Data

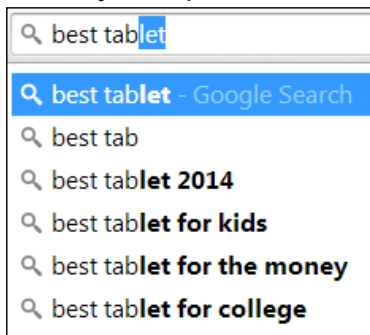


The task was to categorize 800,000 queries into 67 predefined categories.



Search Query Mining Apps: Improving Search

Query completion



- Alternative Query Suggestion:
 - Query completion is helpful but different ways of phrasing a query may get better/alternative results
 - Challenge: Discovering semantically related queries

- Are these queries asking the same thing?
 - “budget hotels in New York”
 - “New York hotels on a budget”
- Are these also related?....
 - “economy hotels in Manhattan”
 - “cheap lodging in the big apple”
 - “buy cheap big apples”

Search Query Mining Apps: Improving Search

- Predict Search Result Relevance - more relevant results
- Task assistance - helping the user perform their task (achieve their goal)
- Next Query Suggestion
 - Assume the query is one step in a task, e.g. booking a vacation. What other steps (queries) can we recommend?

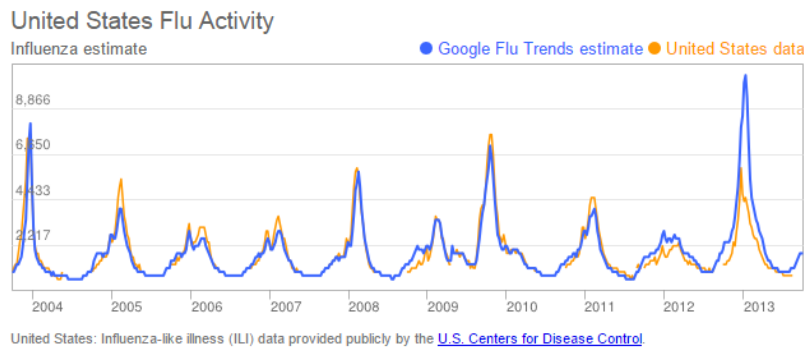


Possible follow-on queries:

- “find cheap hotels in New York”
- “do sightseeing in New York”
- “find New York Restaurants”
- “using the New York subway”
- “how to tip in the US”

Search Query Mining Apps: Others

- Spam page detection ~ detect pages with weak query correlation
- Trend tracking ~ Google trends, Flu track prediction etc.
- Sponsored Search ~ Selecting the best ad for a search query
- User profiling, interest discovery ~ personalisation, ad targeting



Query Suggestion: Possible Approaches

- Classify the query into predefined categories and offer predefined queries (or advice) relevant to that category (e.g. booking a holiday, buying a camera)
 - Text categorisation methods
- Cluster queries based on their content, suggest other queries in the same cluster.
 - Requires a distance (similarity) measure between queries
- Look for queries made together in the same search session
 - Association and Sequence Mining
- Look for queries that lead users to the same landing pages
 - Query-Click Graph Mining

Semantic
analysis,
Text mining

Usage
pattern
mining

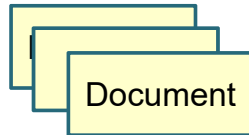
Query Similarity/Query Categorisation

- Using only the words in the query is hard
 - Search queries contain few words and are often ambiguous
 - "cheap hotels in New York" and "budget lodging in the big apple" are very similar queries – but they have no words in common

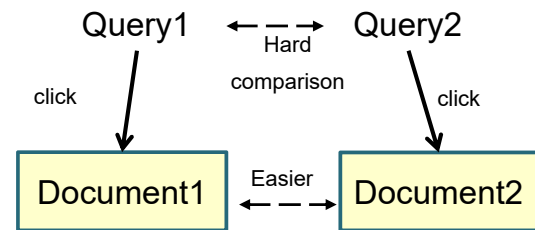
- Possible Approaches

(1) Use a document library* - count how often the query text occurs in the same document.

- $X \sim$ occurrences of Q1 in all docs
- $Y \sim$ occurrences of Q2 in all docs
- $Z \sim$ occurrences of Q1, Q2 in same doc
- Similarity (Q1,Q2) = $Z/\text{SQRT}(X * Y)$ (Ochiai distance)



(2) Analyse the top landing page for the query(s) – much more text to mine and can use document mining methods



Now we can use vector-space/bag of words

Mining the Query-Click Log

- Query Specificity

- The number of distinct documents clicked from a query is an indicator of the queries diversity. More documents implies a diverse (less specific) query.

- Query Attractiveness

- The number of queries leading to a webpage ~ an indication of the webpages "query attractiveness". Pages with high query attractiveness are possible spam pages

- Click Entropy

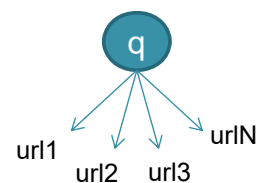
- A query has low click entropy if most users making the query click on the same result – this may indicate they all had similar intent. Queries with high click entropy may indicate the query is ambiguous – can mean different things to different users

Click entropy for a query, $CE(q)$, is calculated as:

$$CE(q) = - \sum_{u \in P_c(q)} p_c(u|q) * \log p_c(u|q),$$

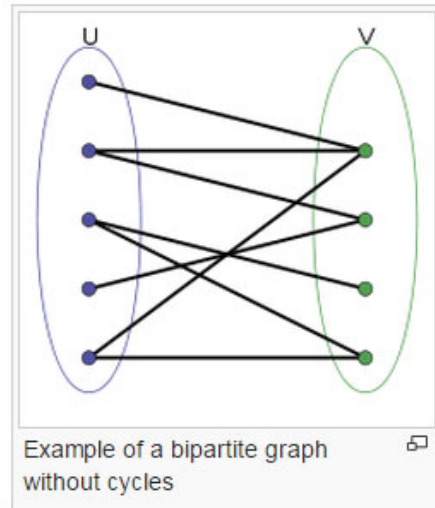
$P_c(q)$ is the collection of URLs clicked on for query q

$p_c(u|q)$ is the percentage of clicks on URL u among all clicks for query q



Mining the Query-Click Log

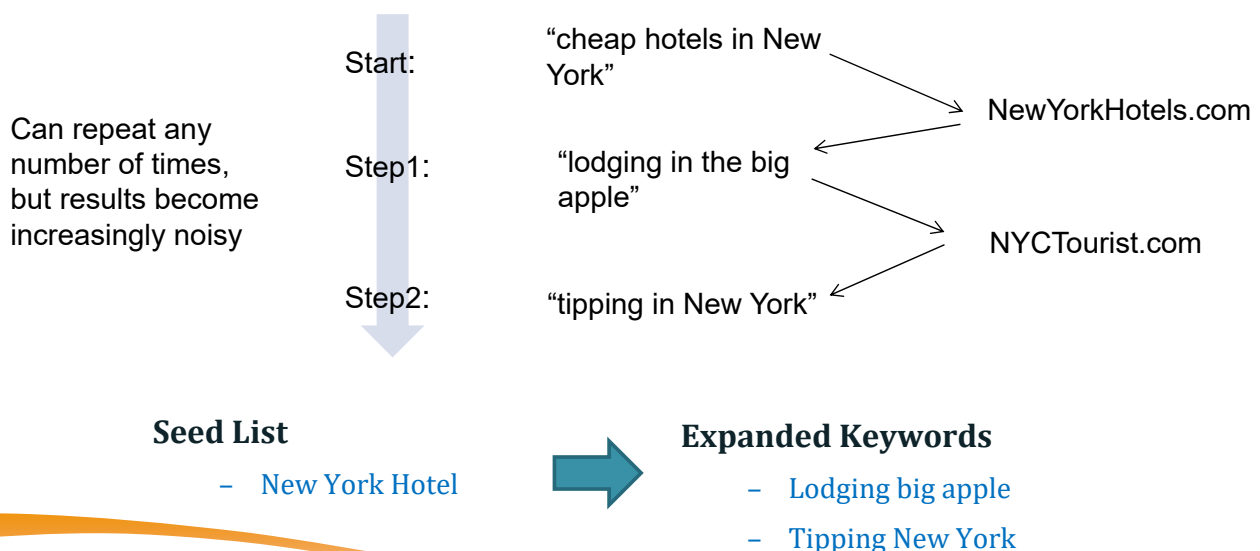
- The Query-click log can be represented as a bipartite graph: nodes are queries and webpages (documents) and the links are clicks
- Issues: documents that are clicked but not relevant constitute noise!



a **bipartite graph** is a graph whose vertices can be divided into two disjoint sets (U & V) and such that every edge connects a vertex in U to one in V

Example: Keyword suggestion for paid search

- Simple graph mining example.
 - Start with a seed list of search keywords and broad match against query logs
 - Follow click links to new queries and extract new keywords.



Query-Click Log Mining Examples

1. Query or Keyword suggestion for paid search
2. Search Result Relevance – get better search engine query results
3. E-Retail – recommending products based on user searches
4. Digital Advertising - predicting clicks on paid search ads

Search Result Relevance

Yandex

Search

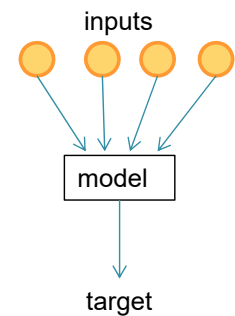
**Yandex is a
large Russian
search engine*

- Yandex Challenge (2012)
 - Predict the 10 most relevant URL's for a search query using past search logs
- The competition search log had 340 million user actions (16.4GB)
 - 30.7M unique queries, 117M unique url's, 43.9M search sessions
- Each record in the log was either:
 - Search query, time since session start, region, ranked list of 10 url's shown
 - Clicked url , time since session start
 - *Commercial intent queries were filtered out. Data was anonymized: sessions, queries and URL's were referenced by **numeric identifiers**, time was given in undisclosed units*
- Prediction target: human experts assigned relevance (0,1) to a *small number* of “query-region-URL” combinations.
 - In a later challenge, dwell time on landing page was used to indicate relevance

Yandex Challenge: Winning Solutions

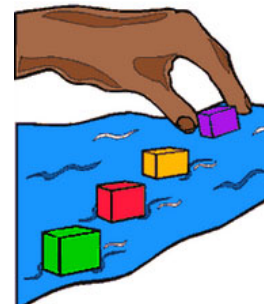
- Prediction Modeling: Inputs were mostly statistical measurements*

- Query measures (#searches, #clicks, CTR, av. click pos...)
- URL measures (#impressions, #clicks, CTR, ..)
- Query->URL measures (#impressions, #clicks, CTR, av. click pos...)
- Positional – URL's near the top often get higher clicks
- Time-based (av. time to click, time to first click, time to last click,...)
 - often the best URL is the last one clicked in a session
- * Queries and URLs were randomised (replaced by index values) hence text mining measures such as TF, IDF could not be used



- The top 10 URLs had to be ranked to win!

- Predict difference in relevance between pairs of URLs then swap pairs around to optimise an overall loss function (LR)
- Used RankNet and LambdaRank(LR)



Bing's search is said to be powered by the RankNet algorithm, which was invented at Microsoft Research in 2005 (wikipedia)

E-Retail: BestBuy Challenge



- Predict which products a (mobile) visitor will be most interested in based on their search query. Two years of mobile searches given. A “small-data” sub-challenge looked only at Xbox games

user	sku	category	query	click_time	query_time
'0001cd0d10bb	2032076	abcat0701002	gears of war	22:56.1	21:42.9
'00033dbced6a	9854804	abcat0701002	Gears of war	35:42.2	35:33.2
'00033dbced6a	2670133	abcat0701002	Gears of war	36:08.7	35:33.2
'00033dbced6a	9984142	abcat0701002	Assassin creed	37:23.7	37:00.0
'0007756f01534	2541184	abcat0701002	dead island	15:34.3	15:26.2
'000878e35cb70	3046066	abcat0701002	Rocksmith	44:36.8	44:22.9
'0008b7e06bc30	2977637	abcat0701002	Nba n2k	10:06.5	09:33.5
'0008f35cccf771	2670133	abcat0701002	Call of duty	05:40.9	05:06.4
'000c22a204a92	9328943	abcat0701002	rock band	02:54.2	02:44.2
'000f573f22d30	1180104	abcat0701002	xbox	39:51.7	36:40.2

There was no guarantee the click resulted from the query – but the two events are always within 5 minutes.

The stock-keeping unit (item) that the user clicked on

The category the sku belongs to

The search terms that the user entered

The test set contained the same fields except sku (the prediction target)

BestBuy Challenge



- An XML file containing product data is also available

```
<products>
- <product>
  <sku>1004622</sku>
  <productId>1218207306495</productId>
  <name>Sniper: Ghost Warrior - Xbox 360</name>
  <source>bestbuy</source>
  <type>Game</type>
  <startDate>2010-05-17</startDate>
  <new>false</new>
  <active>true</active>
  <activeUpdateDate>2012-07-03T04:13:57</activeUpdateDate>
  <regularPrice>19.99</regularPrice>
  <salePrice>19.99</salePrice>
  <onSale>false</onSale>
  <details/>
  <includedItemList/>
  <features>
    <feature>Stalk through the realistically reproduced jungles of South America made with the innovative Chrome engine as you engage in real-time assault</feature>
    <feature>Work with your spotter for better performance and control your breathing for greater accuracy while shooting</feature>
    <feature>Bullet Cam mode allows you to watch bullets strike with pin-point accuracy</feature>
    <feature>Accuracy is affected by factors such as wind and distance falloff for true-to-life ballistics</feature>
    <feature>Advanced animations draw you in as this heart-pounding journey explains the rise to power of a ruthless dictator</feature>
    <feature>Use a full arsenal of weaponry including claymore mines, C4 charges and throwing knives to achieve your objectives</feature>
    <feature>Sniper vs. sniper, real-time tactical assault and fixed machine gun combat scenarios test your skills in various situations</feature>
  </features>
  <offers>
    - <offer>
      <id>promo221700050007</id>
      <heading/>
      <text>Free $10 Savings Code: Find Out How</text>
      <url>http://www.bestbuy.com/site/null/Special Offer/pcmcat276600050013.c?id=pcmcat276600050013</url>
      <imageUrl/>
      <type>special_offer</type>
      <startDate/>
      <endDate/>
    </offer>
  </offers>
</product>
</products>
```

BestBuy: Solutions Tried



- Simplest Solution ~ frequency counting
 - Count the *query->product* instances
 - Build a dictionary:
key=query, values = (sku, clickcnt) pairs
 - Recommend the sku with most clicks
 - Recommend most popular sku's in each product category if no past query->sku mapping exists
- Prediction Model Approach
 - Use TF-IDF to extract features from query and products
 - Use k-NN (or other predictive model) to map query to product

```
'forzasteeringwheel': {'2078113':1},
'finalfantasy13': {'9461183':3, '3519923':2},
'guitarps3': {'2633103':1}
```

E-Retail: CrowdFlower Challenge

- Predict search result relevance for product searches on E-Retail sites

Query

Understand Intent:

The search term is:

laptop lenovo

Google Search for laptop lenovo

Result


Product Title:

Lenovo ThinkPadT420 Intel Corei5 2.5GHz 4GB 750GB 14in Wi-Fi DVDRW CAM Windows 7 Professional (64-bit) (Refurbished)

\$354.99

Result

Product Image:



Product Page

How well does this result match the query?

☐ Off Topic
 ☐ Acceptable
 ☐ Good
 ☐ Excellent

- The intent of the query was not matched
- The results are irrelevant to the search query

- The intent of the query is poorly matched
- The result is somewhat related to the query, but it not a good match

- Matches most of the query intent - or the most important part of the query
- Technically, all parts of the intent are satisfied but result doesn't provide a full, clear and complete answer to the search.

- The query intent is clearly satisfied. This is exactly the product I was looking for
- Result is high quality
- Specifics of the Query appear in the Result

Crowd Sourcing: 261 queries were generated. CrowdFlower assembled a list of products and corresponding search terms. Each rater was asked to give a score of 1,2,3,4 to indicate how well the product satisfied the query

Most important feature ~ distance between the search query and the product title/description, measured using features such as

- Word counts
- TF-IDF
- Jaccard Index

Training Data Fields:

id: Product id
query: Search term used
product_description
median_relevance: Median score by 3 raters
relevance_variance: Variance of the raters scores

Test set was the same but with relevance scores missing

Handling T/F Ratings: Jaccard Index

- Measures the similarity (or distance) between two sets
- Good for the similarity between binary vectors (T/F; Like/Dislike etc.)

$$\text{Sim}_{\text{jaccard}}(A,B) = |A \cap B| / |A \cup B|$$

User	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
U1	1	1	1				1			
U2		1			1	1	1	1		

$$\text{E.g. } \text{Sim}_{\text{jaccard}}(U1, U2) = 2 / 7$$

- As a distance measure:

$$J_{\delta}(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

Paid Search: Predicting Ad Clicks

- Many paid ads can get shown in response to search queries
- Predicting CTR helps search engines set their base CPC (cost-per-click) rates
- CTR depends on many factors including the relevance of ad to the query and the position of the ad on the page

The image shows a Google search interface for the query "uk flights to singapore". It displays a list of sponsored advertisements at the top, followed by organic search results. Annotations include:

- Sponsored ads:** Points to the top section of the search results.
- Organic search results:** Points to the section below the sponsored ads.
- Prime position:** Points to the top-most sponsored ad.
- One ad impression:** Points to a specific sponsored ad.

KDDCup-2012: Predicting Paid Search Clicks

Given training instances (155 million) derived from session logs of the Tencent (chinese lang.) search engine, *soso.com*, predict the CTR of ads in the test instances

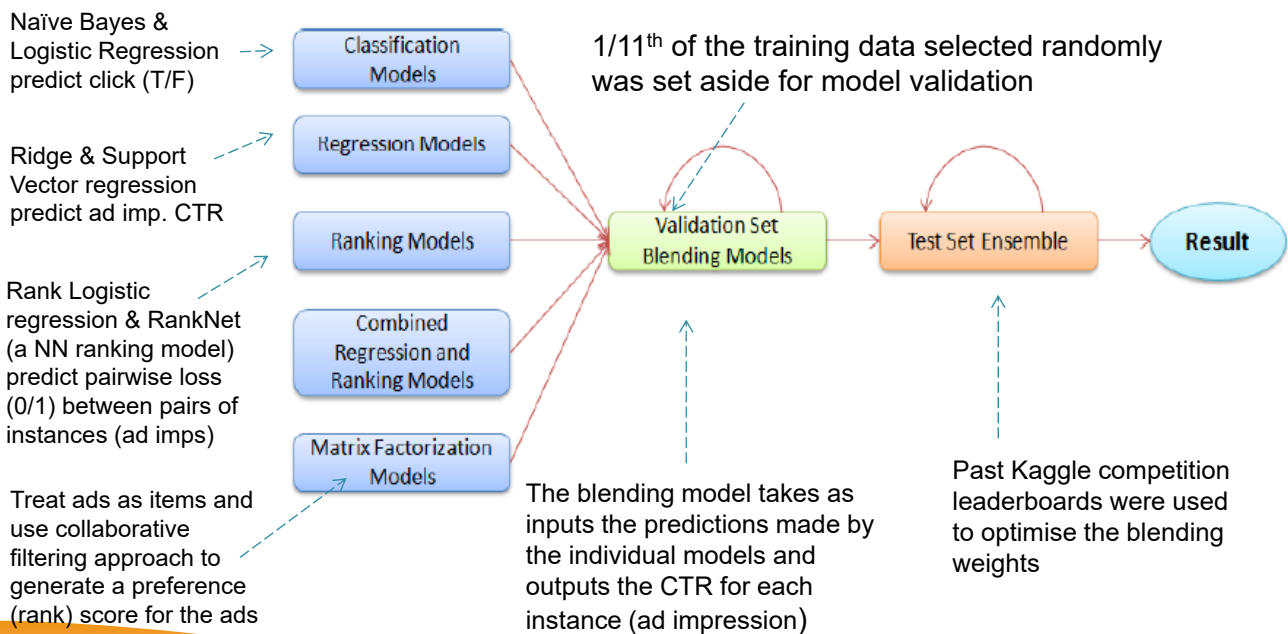
User ID	
Ad ID	
Query ID	an index to a file containing the actual query texts
Depth	number of ads displayed
Position	the order of the ad in the displayed ads
Impressions	number of searches where ad was displayed to user who issued the query
Clicks	number of times, among above impressions, that the user clicked the ad
Display URL	Usually the shortened ad landing page URL (hashed for anonymity)
Advertiser ID	Some advertisers optimise their ads more than others
Keyword ID	Index into a text file (list of keywords bought by advertiser)
Title ID	Index into a text file (title of the ad)
Description ID	Index into a text file (full description of the ad)
Gender	M, F, U
Age	6 age bands (<12, 13-18, 19-24, 25-30, 30-40, >40)

Search instances with same userid, query, adid, depth and position were aggregated to condense the dataset

Impressions & clicks were missing in the test data

KDDCup-2012: Winning Solution

- A two stage ensemble of diverse models (National Taiwan University)



KDDCup-2012: Winning Solution

- Extensive Feature Engineering yielded these model inputs:

Categorical Features

ID	Feature Description
cat_User	categorical feature for UserID
cat_Query	categorical feature for QueryID
cat_Ad	categorical feature for AdID
cat_Keyword	categorical feature for KeywordID
cat_Position	categorical feature for ad's position

Raw Value Features

ID	Feature Description
value_Title	value of TitleID
value_Query	value of QueryID
value_Keyword	value of KeywordID
value_Description	value of DescriptionID
value_User	value of UserID

Grouped ID Value Features

ID	Feature Description
group_Ad	quantized AdID
group_Advertiser	quantized AdvertiserID
group_Title	quantized TitleID
group_Query	quantized QueryID
group_Keyword	quantized KeywordID
group_Description	quantized DescriptionID
group_User	quantized UserID

Basic Sparse Features

ID	Feature Description
binary_Ad	binary expanded AdID
binary_Advertiser	binary expanded AdvertiserID
binary_Query	binary expanded QueryID
binary_Keyword	binary expanded KeywordID
binary_Title	binary expanded TitleID
binary_Description	binary expanded DescriptionID
binary_User	binary expanded UserID
binary_Url	binary expanded Display Url
binary_Gender	binary expanded user's gender
binary_Age	binary expanded user's age
binary_Position	binary expanded ad's position
binary_Depth	binary expanded session's depth
binary_PositionDepth	binary expanded (position,depth)
binary_QueryTokens	binary expanded query's tokens
binary_TitleTokens	binary expanded title's tokens
binary_DescriptionTokens	binary expanded description's tokens
binary_KeywordTokens	binary expanded keyword's tokens

Click-Through Rate Features

ID	Feature Description
aCTR_Ad	average click-through rate of AdID
aCTR_Advertiser	average click-through rate of AdvertiserID
aCTR_Depth	average click-through rate of session's depth
aCTR_Position	average click-through rate of ad's position
aCTR_RPosition	average click-through rate of relative position (depth-position)/depth
pCTR_Ad	pseudo click-through rate of AdID
pCTR_Advertiser	pseudo click-through rate of AdvertiserID
pCTR_Query	pseudo click-through rate of QueryID
pCTR_Title	pseudo click-through rate of TitleID
pCTR_Description	pseudo click-through rate of DescriptionID
pCTR_User	pseudo click-through rate of UserID
pCTR_Keyword	pseudo click-through rate of KeywordID
pCTR_Url	pseudo click-through rate of Display Url
pCTR_RPosition	pseudo click-through rate of relative position (depth-position)/depth
pCTR_Gender	pseudo click-through rate of user's gender
pCTR_Age	pseudo click-through rate of user's age

KDDCup-2012: Winning Solution

Other Numerical Features

ID	Feature Description
num_Position	Numerical value of ad's position
num_Depth	Numerical value of ad's position
num_RPosition	Numerical value of relative position (depth-position)/depth
num_Query	Number of tokens in query
num_Title	Number of tokens in title
num_Keyword	Number of tokens in keyword
num_Description	Number of tokens in description
num_idf_Query	Sum of tokens' idf values in query
num_idf_Title	Sum of tokens' idf values in title
num_idf_Keyword	Sum of tokens' idf value in keyword
num_idf_Description	Sum of tokens' idf value in description
num_Imp_Ad	Number of impressions for AdID
num_Imp_Advertiser	Number of impressions for Advertiser
num_Imp_Depth	Number of impressions for session's depth
num_Imp_Position	Number of impressions for ad's position
num_Imp_Rposition	Number of impressions for relative position (depth-position)/depth

Basic text mining features (IDF = inverse document frequency)

Similarity Features

ID	Feature Description
similarity_tfidf	similarity computed by tf-idf vector between query, title, description and keyword (6 features)
similarity_topic_6	Topic similarity between query, title, description and keyword (6 features, from 6 topics LDA)
similarity_topic_20	Topic similarity between query, title, description and keyword (6 features, from 20 topics LDA)

Similarities between user query and ad description/text

KDDCup-2012: Winning Solution

Model	Validation AUC	Public Test AUC
Naïve Bayes	0.8108	0.7760
Logistic Regression	0.8152	0.7888
Ridge Regression	0.7539	0.7351
SVR	0.8135	0.7705
RLR	0.7442	0.7220
RankNet	0.7941	0.7577
CRR	0.8174	0.7818
Regression Based MF	0.8077	0.7775
Ranking Based MF	0.8246	0.7968

Individual Model Test Scores

Model	Public Test AUC
1-level SVR	0.8038
1-level LambdaMart	0.8051
1-level RankNet	0.8058
2-level SVR	0.8031
2-level CRR	0.8050
2-level RF-LambdaMart	0.8059
2-level RankNet	0.8062

Different Blending Models Tried

AUC = area under the ROC curve

Day Agenda

- Web Server Logs ~ structure and issues
- Web Server Log mining
- Search Query Log mining
- Behavioral targeting and Ad-click Prediction ←
- Assignment1

User Behavioral Modeling

- What can we learn about a user by looking at their recent browsing & search behavior?
- **Behavioral Targeting (BT)** - Uses information collected from an individual's web-browsing behavior to select advertisements to display. Used by online website publishers and advertisers. (Wikipedia)



Other data sources such as demographics, geo-location, and past purchases are also used to enhance personalization.



GENDER:
Female



LOCATION:
Illinois



PAST PURCHASES:
Inline skates



Other data sources
can be used when
available

Behavioral Targeting using Segmentation

- Assign user to commercial intent (in-market) segments based on their past browsing behavior on the ad network. Normally done off-line (e.g. overnight in batch mode)
- When the user revisits any website on the display network they get shown ads relevant to the segments they are in

Commercial Intent Segments:

Camera buyer, home buyer, car buyer,...

Lifestyle Segments:

Sports enthusiast, news reader, health conscious,...

Demographic Segments:

Young Males, Retirees,...

Usually built manually by creating lists of queries, webpages etc. Users visiting these pages or doing those queries get assigned to the segments.

Camera-queries:

compact camera
SLR
zoom lens
Nikon
Canon
etc....

Camera-pages:

www.cameras.com
www.camera-reviews.com
www.bestbuy/cameras/*
etc....

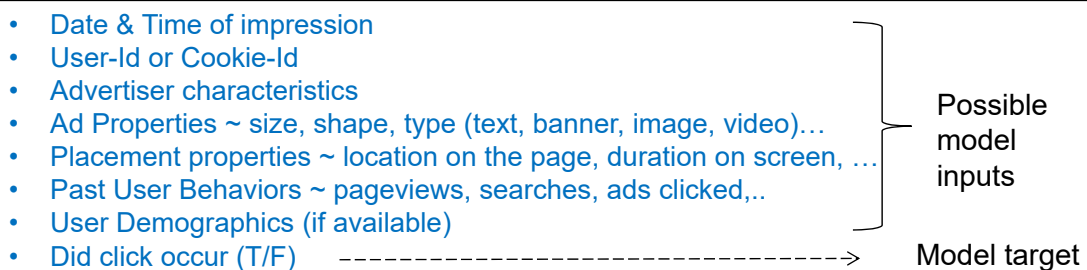
wildcard

Sometimes rules are used to combine the lists:

e.g. IF (camera-queries > 1) OR (camera-pages > 2)
Then segment = camera-buyer

Display-Ad Click Prediction

- Use ***past user behaviors + current context + ad properties*** to predict clicks on specific ads or ad categories. Can execute in real-time to help Ad selection when a user visits a website in the ad network
- Each past Ad impression is a potential training example:



- Logistic Regression is commonly used due to its scalability

*See this paper for a detailed description of building display ad click models

Simple and scalable response prediction for display advertising

OLIVIER CHAPELLE, Criteo[†]
EREN MANAVOGLU, Microsoft
ROMER ROSALES, LinkedIn

<http://people.csail.mit.edu/romer/papers/TISTRespPredAds.pdf>

Display-Ad Click Prediction: Issues

- How to turn user behaviors into suitable inputs for a prediction model?
- E.g. Past events for one user:

- ☐ Searched for “cheap flights to New York”, 11.15am, Jan2nd
- ☐ Clicked on “Travelocity” link in search results, 11.16am, Jan2nd
- ☐ Viewed <http://www.straitstimes.com/breaking-news/money/story/ps4-and-xbox-fuel-sales-20140117>, 10pm, Jan3rd
- ☐ Received Ad2317 (SingTel) 10.00pm, Jan 3d
- ☐ Searched for “tourist hotels in New York”, 8.10pm, Jan4th
- ☐ Clicked on “Holiday Inn” link in search results, 8.12pm, Jan4th
- ☐ Clicked on Ad2523 (discount offer from Hilton), 8.15pm, Jan 4th
- ☐ Received email from Target.com, 9am, Jan 5th
- ☐ Viewed <http://www.amazon.com/gp/goldbox/cell-phones> , 2.32pm, Jan5th
- ☐ Viewed <http://www.amazon.com/categories/scifi-books/id3456>, 2.32pm, Jan5th

- How many days or weeks to look-back?
 - When does behavior go stale?
(e.g. interest in hotels in NY probably expires after a trip)

Encoding Behaviors: One Method...

- Categorise the user events (page-views, searches...) into a topic taxonomy

Clothing & Shoes
Clothing & Shoes/Baby Clothing & Accessories
Clothing & Shoes/Boys Clothing & Accessories
Clothing & Shoes/Costumes
Clothing & Shoes/Girls Clothing & Accessories
Clothing & Shoes/Girls Clothing & Accessories/Formal Wear
Clothing & Shoes/Gloves & Mittens
Clothing & Shoes/Hosiery & Socks

- Taxonomies can have many thousands of entries (nodes)
- Typically there is an accuracy/granularity versus computation trade-off

- Count the user events in each category in (say) the last 14 days
 - e.g. User2341, Clothing/Babywear=1, Clothing/Maternity=2,... etc.
- Combine with all other features, one record per ad impression

Possible model inputs

Impression	User	Demographics			Recent Behaviors			Advertiser and Ad Details			Target
Date-Time	ID	Sex	Age	...	Node1	Node2	...	Size	Type	...	Clicked
1/3/15:07:15	1	M	31	...	0	1	...	200*600	image	...	T
5/8/15:16:15	2	F	18	...	2	3	...	400*300	banner	...	F

But this can result in records with huge dimensionality!

...or, using a Bag-of-Words Approach

- Bag of words approach eliminates the need for the event categorisation

Impression	User	Demographics			Recent Behaviors			Advertiser and Ad Details			Target
Date-Time	ID	Sex	Age	...	Term1	Term2	...	Size	Type	...	Clicked
1/3/15:07:15	1	M	31	...	0	1	...	200*600	image	...	T
5/8/15:16:15	2	F	18	...	2	3	...	400*300	banner	...	F

E.g. the terms extracted from all of the users recent page-views and search queries

budget	cheap	chicago	hotel	motel	new york
1	0	0	1	0	1	
0	2	1	0	1	0	
1	1	1	0	1	1	

But this can result in records with even larger dimensionality!

Encoding Behaviors : Feature Hashing*

- Can reduce feature dimensionality significantly
- Decide on a vector length (the hash table size) and then convert each feature into hash values. Popular for document categorisation where the features are the words or n-grams in each document.
- The output vector is then the counts of the individual hash values, e.g.

Hash feature1	Hash feature2	...	Hash feature N	Class value
3	0		1	T
0	2		6	F

```
function hashing_vectorizer(features : array of string, N : integer):  
    x := new vector[N]  
    for f in features:  
        h := hash(f)  
        x[h mod N] += 1  
    return x
```

*Made popular by *Vowpal Wabbit*. An open source, fast, out-of-core learning system library and program developed at Yahoo! Research, and currently at Microsoft Research

Encoding Behaviors : Feature Hashing

- Hashing can reduce the number of terms in a bag of words approach.

	Term1	Term2	Term3	...		Term N (very large)
event1	3	4	1	...		
event2	2	7	0		
event3	0	4	1		
.....						



	Hashval1	Hashval2		Hashval M
event1	5	2	...	
event2	3	1	...	
event3	0	3	...	

Where $M \ll N$

E.g. Hashing user comments

USERTEXT	SENTIMENT
I loved this book	3
I hated this book	1
This book was great	3
I love books	2



Examples of the bigrams...

TERM (bigrams)	FREQUENCY
This book	3
I loved	1
I hated	1
I love	1

+

Examples of the unigrams...

Term (unigrams)	FREQUENCY
book	3
I	3
books	1
was	1



Rating	Hashing feature 1	Hashing feature 2	Hashing feature 3
4	1	1	0
5	0	0	0

If the value in the column is 0, the row did not contains the hashed feature.
If the value is 1, the row did contain the feature.

<https://msdn.microsoft.com/en-us/library/azure/dn906018.aspx>

Feature Hashing of Categorical Variables

- Hashing can also reduce and optimise categorical variables that have large numbers of categories, e.g. a users address or their last 10 searches:

	Address	Search1	Search2	...
user1	hashval	hashval	hashval	
user2	hashval	hashval		
user3	hashval			

Kaggle: Criteo Challenge

Predict click-through rates on display ads

Display advertising is a billion dollar effort and one of the central uses of machine learning on the Internet. However, its data and methods are usually kept under lock and key. In this research competition, Criteo Labs is sharing a week's worth of data for you to develop models predicting ad click-through rate (CTR). Given a user and the page he is visiting, what is the probability that he will click on a given ad?

Training Data:

Label - Target variable that indicates if an ad was clicked (1) or not (0).

I1-I13 - A total of 13 columns of integer features (mostly count features).

C1-C26 - A total of 26 columns of categorical features. The values of these features have been hashed onto 32 bits for anonymization purposes

<https://www.kaggle.com/c/criteo-display-ad-challenge>

Criteo Data: High Dimensionality

The raw data:

Label	I1	I2	...	I13	C1	C2	...	C26
1	3	20	...	2741	68fd1e64	80e26c9b	...	4cf72387
0	7	91	...	1157	3516f6e6	cfc86806	...	796a1a2e
0	12	73	...	1844	05db9164	38a947a1	...	5d93f8ab
					⋮			
?	9	62	...	1457	68fd1e64	cfc86806	...	cf59444f

#Train: $\approx 45\text{M}$
 #Test: $\approx 6\text{M}$
 #Features after one-hot encoding: $\approx 33\text{M}^*$

One-hot encoding turns a categorical feature with N values into N boolean features

Status (single, married, divorced)
 ↓
 Single (0,1), Married (0,1), Divorced(0,1)

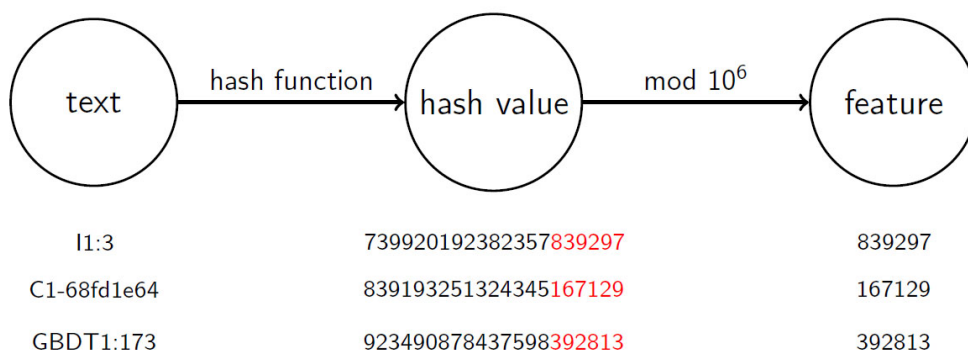
* hashing was to 32bits so the max. number of one-hot coded features = $26 * 2^{32} \sim 112\text{Billion}$

The Winning Solution

3 Idiots' Approach for Display Advertising Challenge

Yu-Chin Juan, Yong Zhuang, and Wei-Sheng Chin
 NTU CSIE MLGroup

- Feature Hashing to Reduce Dimensionality



- Use Ensemble for prediction:
 Logistic Regression +
 Factorisation Machines +
 Gradient Boosted Trees

Method	Public	Private
LR-Poly2	0.44984	0.44954
FM	0.44613	0.44598
FM + GBDT	0.44497	0.44483
FM + GBDT (v2)	0.44474	0.44462
FM + GBDT + calib.	0.44488	0.44479
FM + GBDT + calib. (v2)	0.44461	0.44449

Outbrain Click Prediction



Can you predict which recommended content each user will click?

\$25,000 · 979 teams · 9 months ago

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#)

The internet is a stimulating treasure trove of possibility. Every day we stumble on news stories relevant to our communities or experience the serendipity of finding an article covering our next travel destination. Outbrain, the web's leading content discovery platform, delivers these moments while we surf our favorite sites.



Currently, Outbrain pairs relevant content with curious readers in about 250 billion personalized recommendations every month across many thousands of sites. In this competition, Kagglers are challenged to predict which pieces of content its global base of users are likely to click on. Improving Outbrain's recommendation algorithm will mean more users uncover stories that satisfy their individual tastes.

<https://www.kaggle.com/c/outbrain-click-prediction>



ATA/BA-DAT/webusagemining/V6

© 2018 NUS. All rights reserved.

Page 67

Source: [edition.cnn.com/2016/08/23/middleeast/iraq-nineveh-mosul-scene/index.html](#) Publisher Document

or quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

CNN Regions » Battle looming: Iraqi troops, militia inch towards ISIS-held Mosul

"I am so happy for them," the man said. "But I am heartbroken myself. My parents were not able to come with me. I don't know how I am going to get them out."

Paid Content Recommended by Outbrain

Promoted Content Set

- Mapping the Startup Nation: The 12 most popular Tech Hubs in...
Viola Notes
- First time in Israel: Business degrees in Ramat Gan and New...
Israel News
- The most addictive game of the year! Play with 15 million Players...
Forge Of Empires
- How to Avoid Everyday Pain Landmines
Womens Health
- How One Brand is Disrupting the \$63 Billion Makeup Industry
The Huffington Post
- Find out what special ingredient makes this omelette so tasty
HomeMadebyYou

Promoted Content Item

Goal is to rank the recommendations in each group by decreasing predicted likelihood of being clicked.



ATA/BA-DAT/webusagemining/V6

© 2018 NUS. All rights reserved.

Page 68

Outbrain Challenge: Data files

1. Pageview = UserID, docID, datetime, *platform, geolocation, traffic-source*
2. Event = links a pageview to a displayID
3. Click = displayID, adID, clicked(T/F) ← *One click record per ad that's in the display box for a specific pageview. Only one was clicked*
4. Info on the documents
5. Info on the ads

