Web Usage Mining

Web Science Course

Leibniz University Hannover

16 November 2010

Eelco Herder

Outline

Why Web Usage Mining

Crawling and Mining Methods

Models of Online Browsing Behavior

How Do We Browse the Web

Prediction Models

Application: Personalization

Why Web Usage Mining

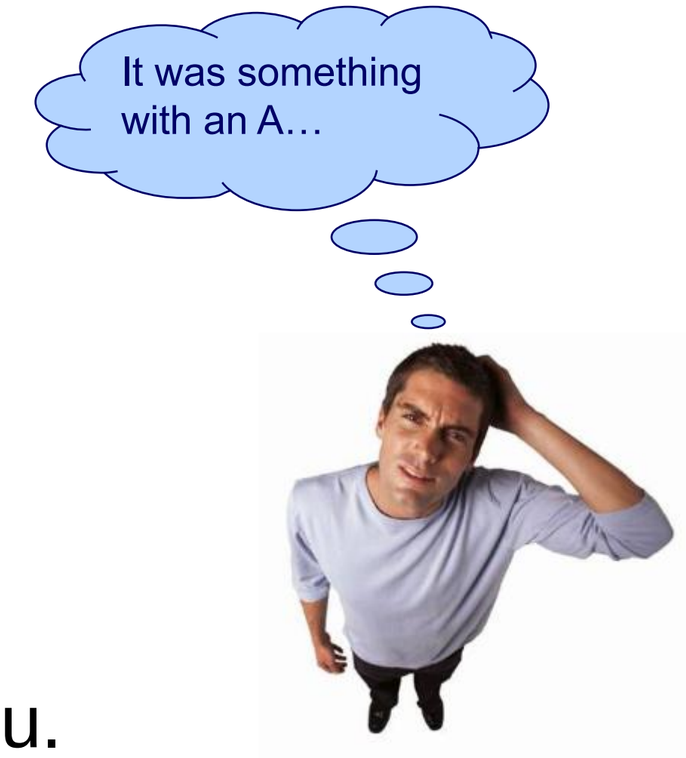
Finding and refinding information on the Web

Have you bookmarked all visited Web sites that offer good bargains, provide handy services, or that keep you informed? Most likely not.

If you are like most people, you just trust that you will find the site again.

However, after a while you probably won't remember the exact name of this great bookstore or how you found it and with which keywords you could find it again.

You might even forget about its existence, unless someone or something reminds you.



What is Web Usage Mining

Web usage mining is about discovering interesting facts in user navigation.

- Frequently reoccurring actions
- Frequently visited pages
- Commonly followed paths
- Usage of search tools
- Strategies used to locate information

Although user tasks differ wildly, many regularities and useful patterns can be discovered.

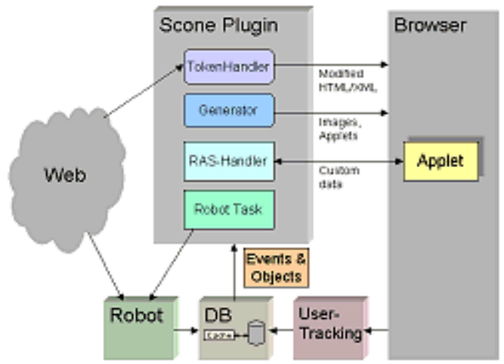
Application areas

- Target potential customers for electronic commerce
- Enhance the quality and delivery of Internet information services to the end user
- Improve Web server system performance
- Identify potential prime advertisement locations
- Facilitates personalization/adaptive sites
- Improve site design
- Fraud/intrusion detection
- Predict user's actions (allows prefetching)

Crawling and Mining Methods

Where does Web usage data come from

Web usage data is collected automatically via software logging tools



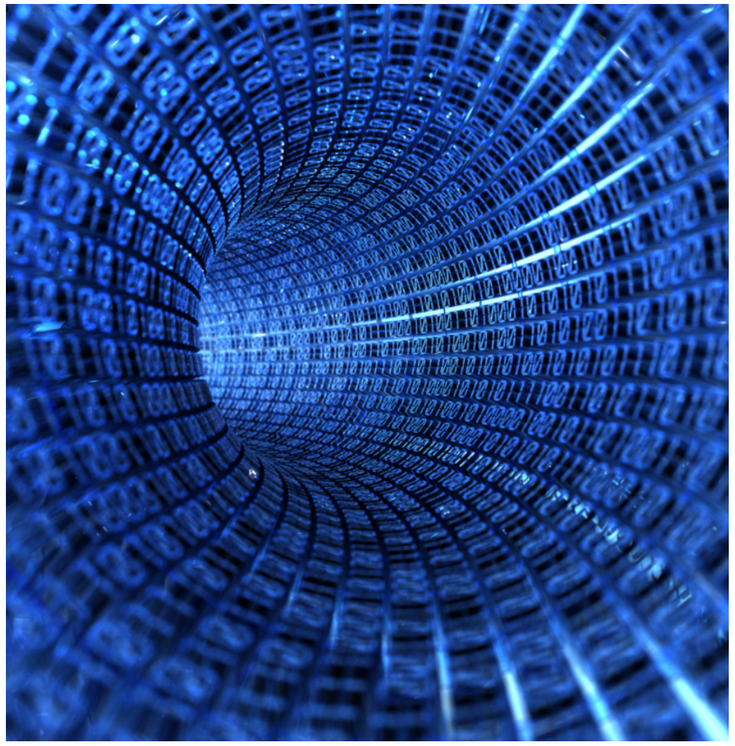
Advantage

No manual supervision required

Disadvantage

Data can be skewed (e.g. due to the presence of robot traffic)

Important to identify robots (also known as crawlers, spiders)



Three sources of data

Web Server

- Data from multiple users on one site
- In general limited to click-behavior

Web Client (i.e. Browser)

- Data from one user on multiple sites
- More sophisticated tracking (e.g. mouse movements)
- Requires user cooperation (e.g. installation of a program)

Proxy Server

Data from multiple users on multiple sites

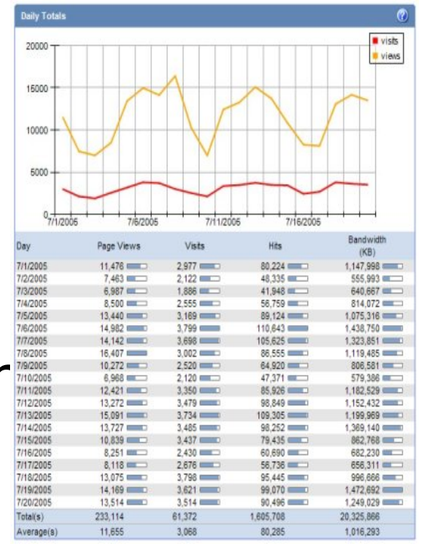
Sits between server and client

Analyses (and possibly modifies) the data stream

Web Server Log Data

This data is automatically gathered at the web server.

It is ‘for free’ if you have access to it.



Day	Page Views	Visits	Hits	Bandwidth (KB)
7/1/2005	11,476	2,977	80,224	1,147,998
7/2/2005	7,463	2,122	48,335	555,993
7/3/2005	6,987	1,886	41,948	640,667
7/4/2005	8,500	2,555	56,759	814,072
7/5/2005	13,440	3,169	89,124	1,075,316
7/6/2005	14,982	3,799	110,643	1,438,750
7/7/2005	14,142	3,698	105,625	1,323,851
7/8/2005	16,407	3,002	86,555	1,119,485
7/9/2005	10,272	2,520	64,920	806,581
7/10/2005	6,968	2,120	47,371	579,386
7/11/2005	12,421	3,350	85,926	1,182,529
7/12/2005	13,272	3,479	98,849	1,152,432
7/13/2005	15,091	3,734	109,305	1,199,969
7/14/2005	13,727	3,485	98,252	1,369,140
7/15/2005	10,839	3,437	79,435	862,768
7/16/2005	8,251	2,430	60,690	682,230
7/17/2005	8,118	2,676	58,736	656,311
7/18/2005	13,075	3,798	95,445	996,666
7/19/2005	14,169	3,621	99,070	1,472,692
7/20/2005	13,514	3,514	90,496	1,249,029
Total(s)	233,114	61,372	1,605,708	20,325,866
Average(s)	11,655	3,068	80,285	1,016,293

Basically it is a list of page requests: ip address, date, comm
client identifier

Data needs to be cleaned:

- Distinguishing between users and sessions
- Path completion (not all page requests arrive at the server due to caching by browsers and proxy servers

Example Server Log

213.6.31.68 - - [01/May/2004:22:38:32 +0200] "GET /forsale.html HTTP/1.1" 200 14956 "http://www.fortepiano.nl/indexforsale.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)“

213.6.31.68 - - [01/May/2004:22:38:34 +0200] "GET /pictures/forsale/bertsche1835-small.jpg HTTP/1.1" 200 5753 "http://www.fortepiano.nl/forsale.html" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"

Web Crawlers / Robots / Spiders

Basically, a web crawler (‘robot’/ ‘spider’) starts at a page and recursively follows all the links.

One can tell a crawler to limit itself to one server, but one doesn’t have to.

Search engines, such as Google, use a crawler to find the information they need.

Some elements that a web crawler can capture:

- Page titles, authors, web addresses
- Page content
- Links between pages

How to recognize robots?

Robot requests identified by classifying page requests using a variety of heuristics

- e.g. some robots self-identify themselves in the server logs

Robots explore the entire website in breadth first fashion

- Periodic Spikes (can overload a server)
- Lower-level constant stream of requests

Humans access web-pages in depth first fashion

- Daily pattern: Monday to Friday
- Hourly pattern: peak around midday, low traffic during nights

Cleaning Web Usage Data

Server side

Recognizing individual users

- User login, cookies, session identifiers, time-out heuristic (25 min.)

Removal of site visits by robots, such as the Google robot that tries to find modified, deleted and new pages

Path completion: inserting page visits that were not captured due to browser caching

Client side

Merge multiple frame visits to one page visit

Removal of page requests to adservers

Removal of artifacts, e.g., reloading pages (e.g. news tickers)

Proxies – Advantages and Disadvantages

Compared to Server-Based

continuity of tracking between different sites

user identification solved by login

no problems due to caching

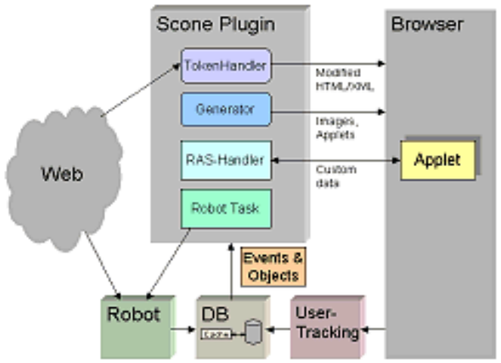
Compared to Client-Based

less flexible than modified browser

more flexible than e.g. Javascript add-ons

applying and distributing changes is easier

user groups can be observed and compared



Models of Online Browsing Behavior

Web pages, links and search

Web sites consist of pages, connected by links.

Menus build a hierarchy of these pages. Menu links provide the context. They reduce cognitive overhead.

Links within text are associative and overrule the site hierarchy.

Site search provides global navigation.

Different types of models

Cognitive: building a mental model on how users perceive and use the web.

Empirical: collecting and analyzing actual user behavior in (laboratory) studies.

Statistical: finding regularities in web user logs.

Typically, a combination of these approaches is used in web usage mining.

Something completely different

Imagine you are a predator looking for prey…

Suppose: plenty easy-to-catch sheep available.

No need to spend energy by moving to another hunting ground.

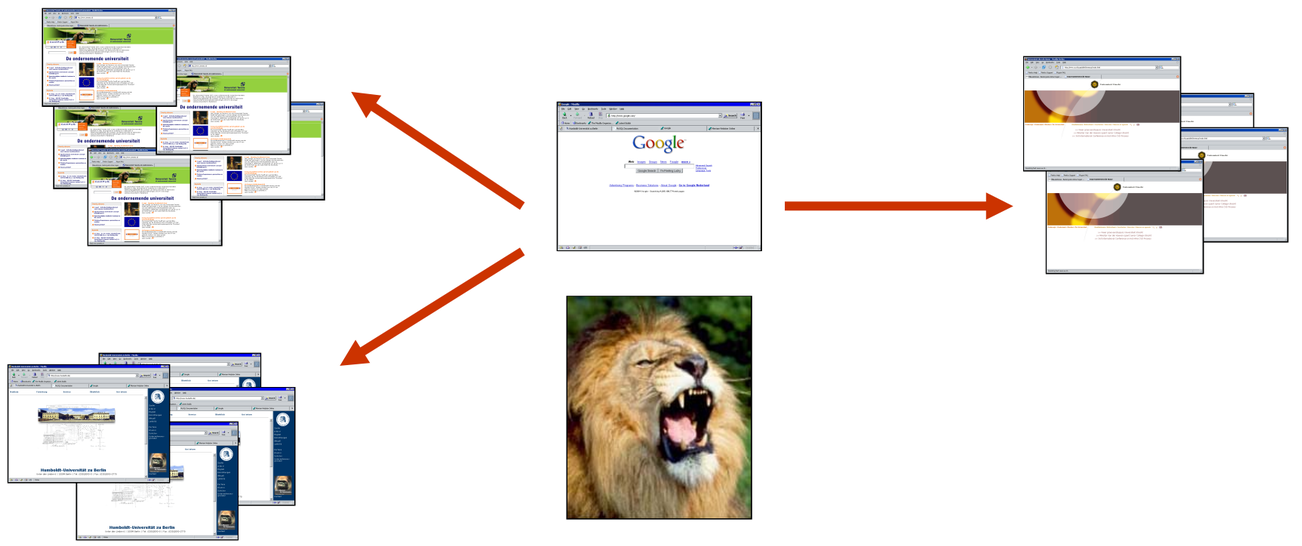
Suppose: only some hard-to-catch deer available

It might be better to move to another hunting ground, although traveling takes time.

Well, the same applies to the Web too

Web sites can be seen as hunting grounds.

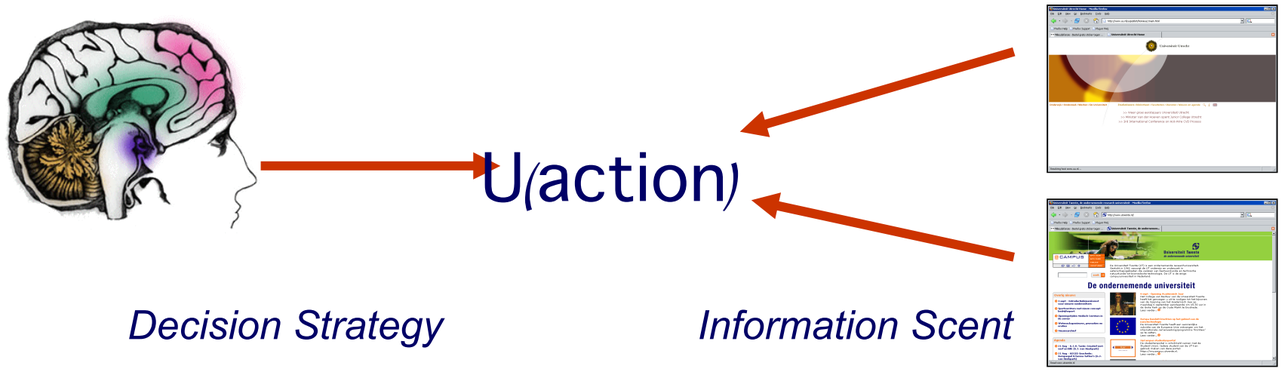
Once you suspect that it might be easier to find the information you need elsewhere, you move to another web site.



Information Foraging

Assumption: users try to maximize their information gain by choosing actions with the highest expected utility.

Based on proximal cues: visible links, menus, search results, …



Maximize the rate of gain

Users want to find as much relevant information as they can get, and to spend as little time and effort as possible.

Time and energy spent can be divided in:

- cost for finding a good site (hunting ground)
- cost for locating the information in the site

$$\text{Rate of Gain} = \frac{\text{Information Gained}}{\text{Time and Energy Spent}}$$

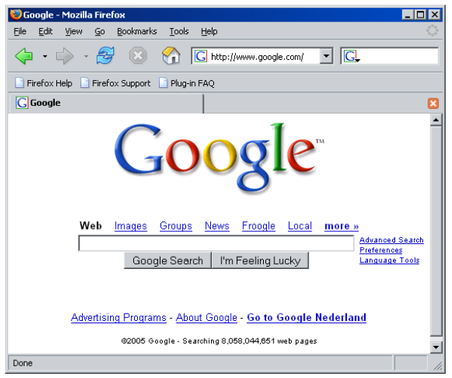
Search engines and information scent

When it is easy to change information patch, it is likely that users will decide sooner to move elsewhere.

Search engines have made it far easier to travel from one information patch to another.

This means a great challenge for site designers to clearly show what their sites offer.

Users typically oversee a big problem associated with site switching: they lose context in their search.

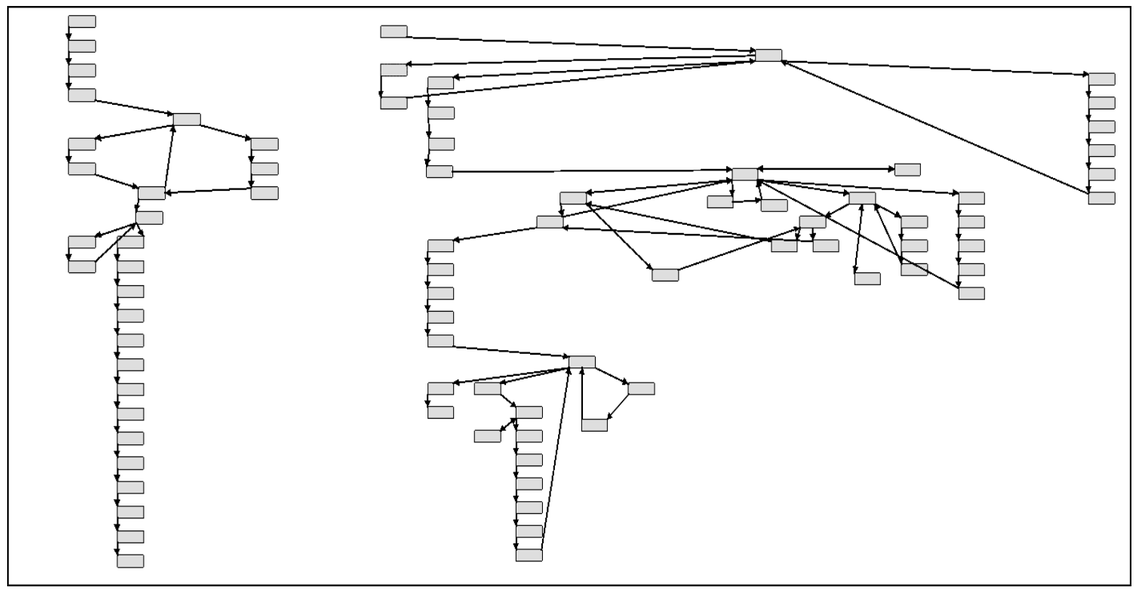


Is backtracking wrong?

Most IF models assume that backtracking indicates failure.

However, backtracking is observed to be a highly effective strategy:

- Learning the way a site is organized
- Making use of navigation hubs, (e.g. index pages)



How do we browse the Web?

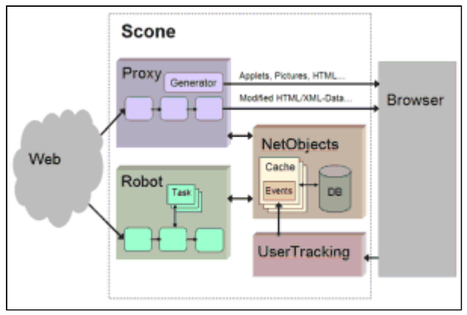
A long-term client-side study

Cooperation with Harald Weinreich, Matthias Mayer and Hartmut Obendorf from Uni Hamburg.

Activities of 25 participants were logged for a period of 3 months on average.

- 17 German participants, 8 Dutch participants
- Mainly academic background, like earlier studies

Activities were logged with the proxy of the Scone framework.

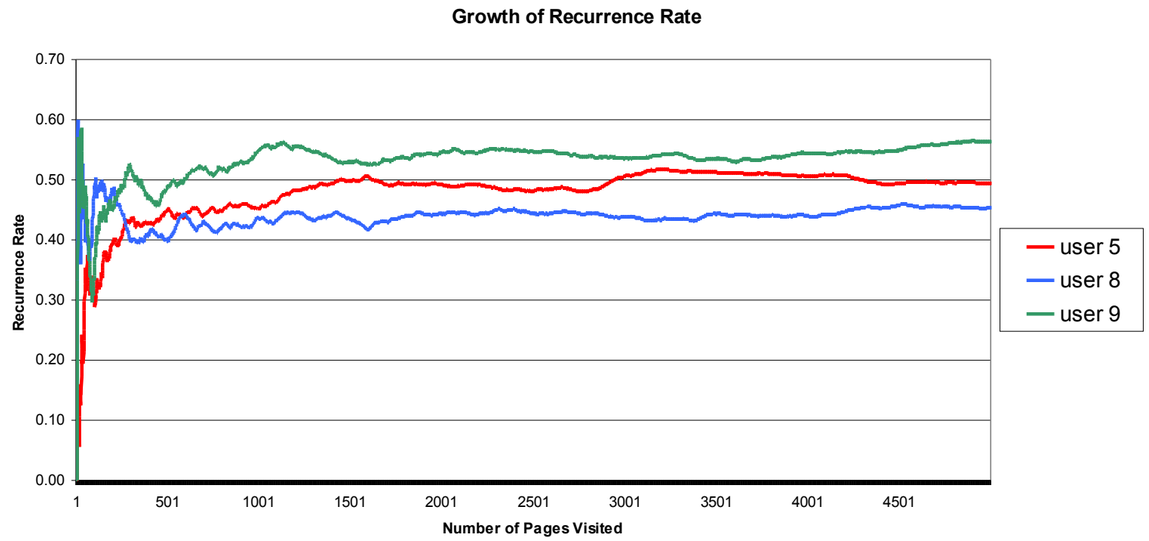


Recurrent behavior on the Web

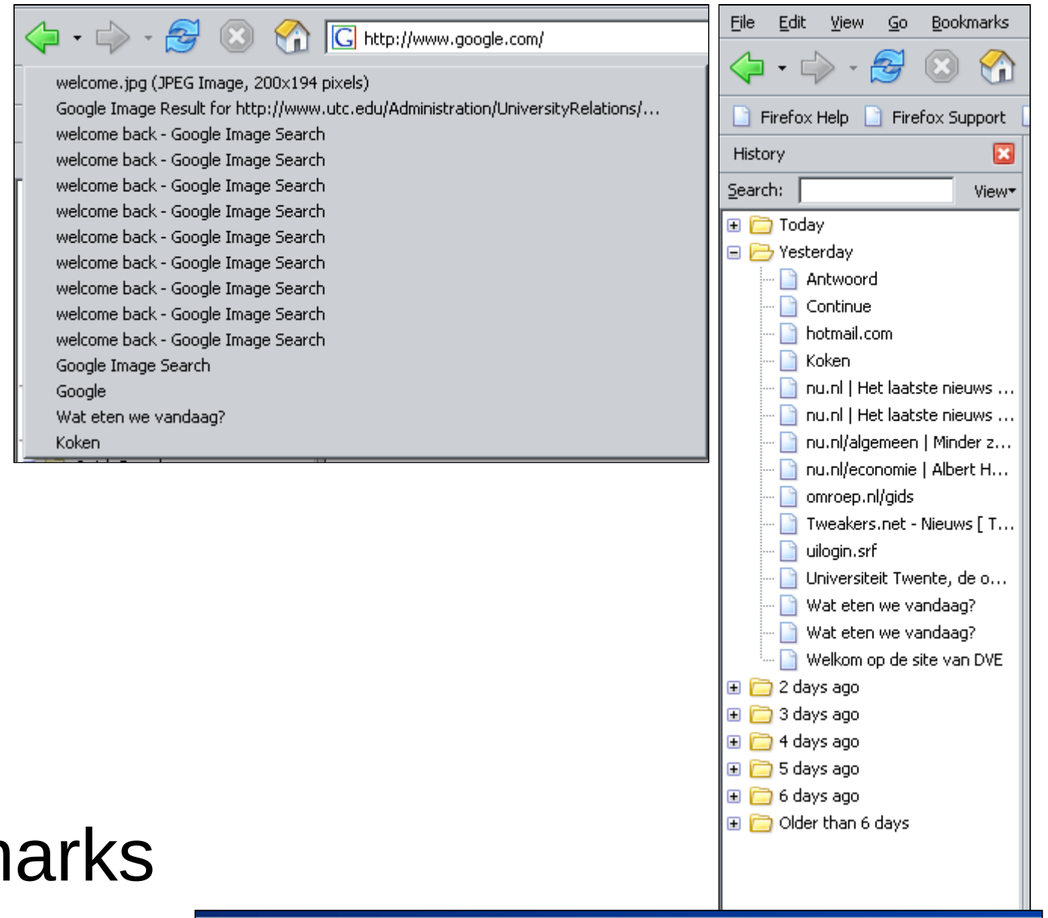
From earlier studies it is known that recurrent behavior is the main activity in web navigation (>50% of all Web activity)

Users revisit pages because

- There might be new content
- They want to further explore the site
- The page has a special purpose
- They are authoring a page
- The page leads to a page they want to revisit



Browsers support revisits



Back Button

Most frequently used tool

For short-term revisits

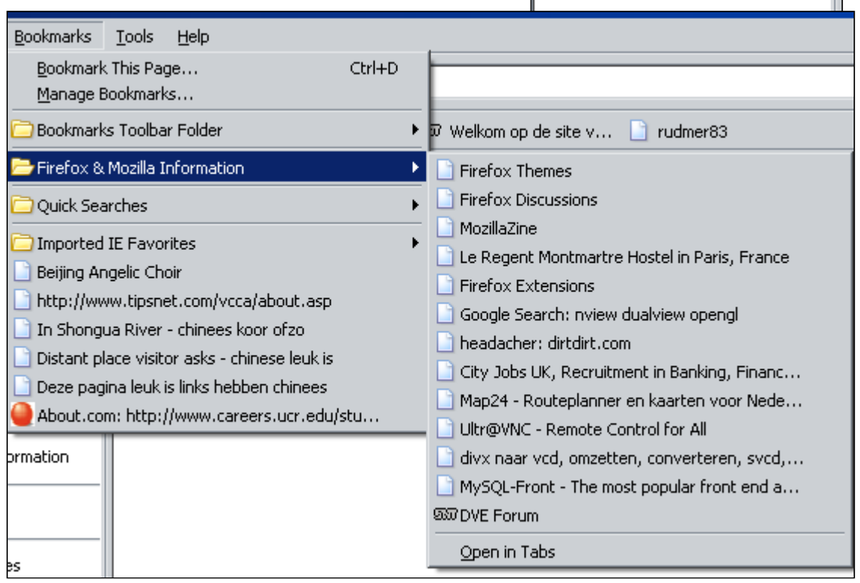
Bookmarks

Users have problems managing bookmarks

List too large or outdated bookmarks

History Lists

Temporal ordering is not very useful



These tools are not integrated

How do we revisit pages?

Short-term revisits

backtracking, undo, reference sites, search engine

back-button

Medium-term revisits

re-utilization, monitoring, forums, educational pages

url auto-completion

Long-term revisits

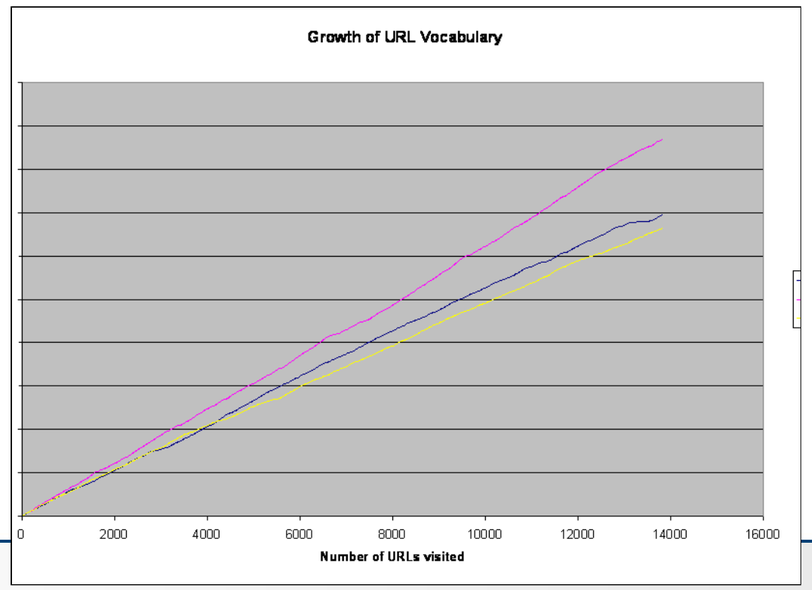
refinding information, re-occurring tasks, travel planning, weekend activities

erm... bookmarks, …. erm… google?

Growth of URL vocabulary

The ratio between the number of pages visited for the first time and the number of revisited pages is about the same through time.

An increased slope of growth indicates that a user has been looking for new information – perhaps started a new task.

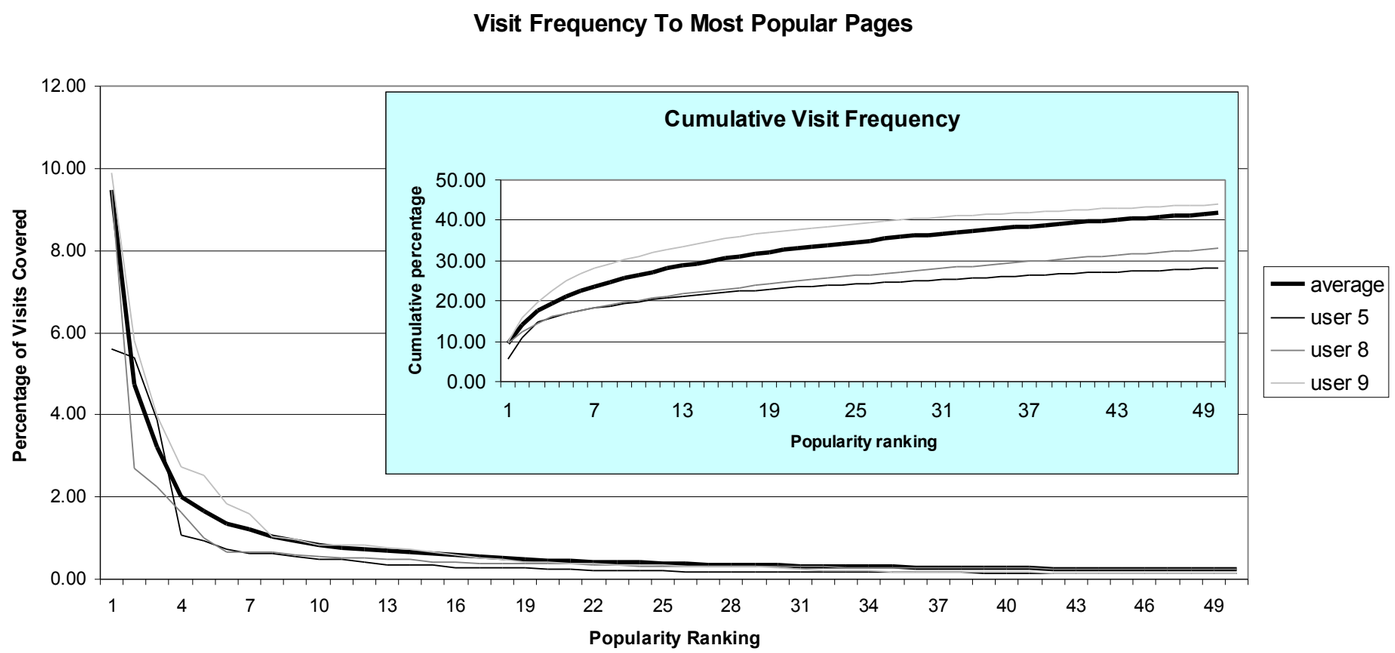


Revisits to popular pages

A small number of pages is visited very frequently, a large number of pages is visited very infrequently.

This distribution is called a power law and can be observed in many cases.

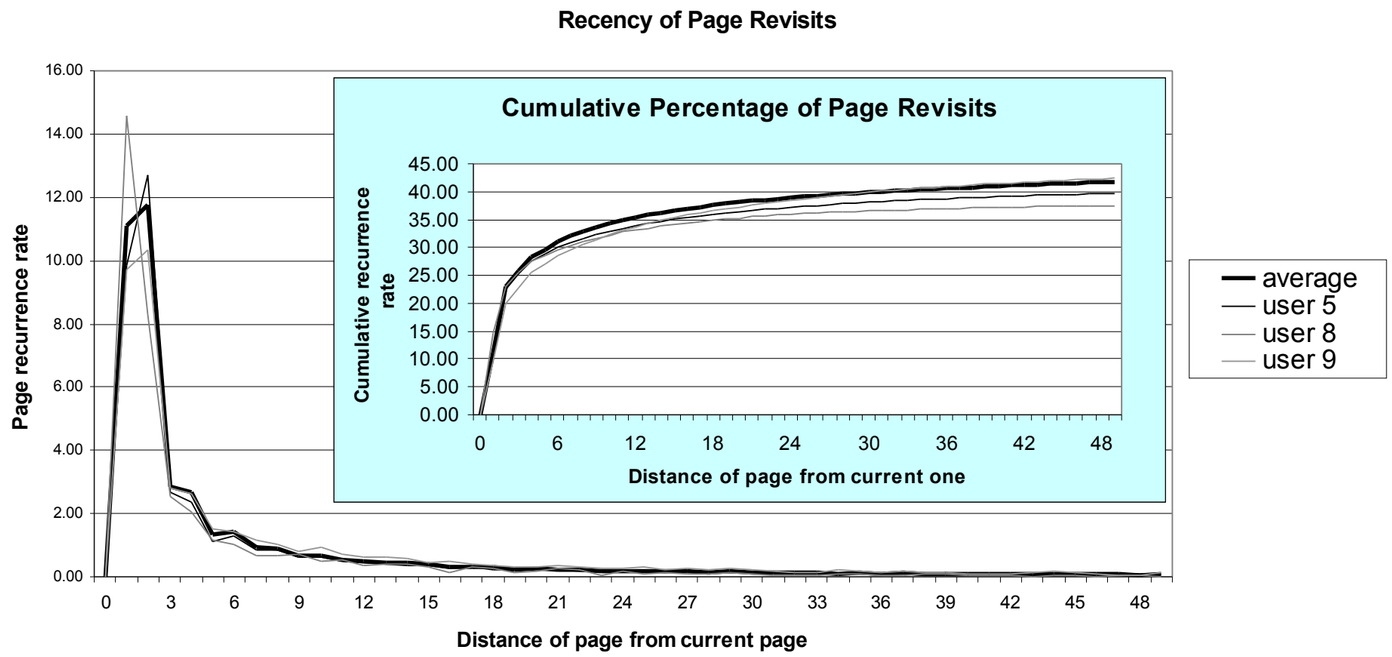
The fifteen most popular page account for 30% of all revisits.



Recency of revisits

Most pages that are revisited have been visited quite recently – the longer the time has passed, the less likely it is that a page will be revisited.

50% of all page revisits have been visited less then four pages before.



Revisits and search

About 12% of all page visits were visits to search engine result pages.

79% of all result pages were followed by a first-time page visit.

In search intensive sessions the amount of revisits is lower.

Users backtrack frequently to result pages: 30% of all visits to the search engine were returns to the result pages.

Backtracking is decreasing

We found dramatically less back button actions than in earlier studies

Two causes:

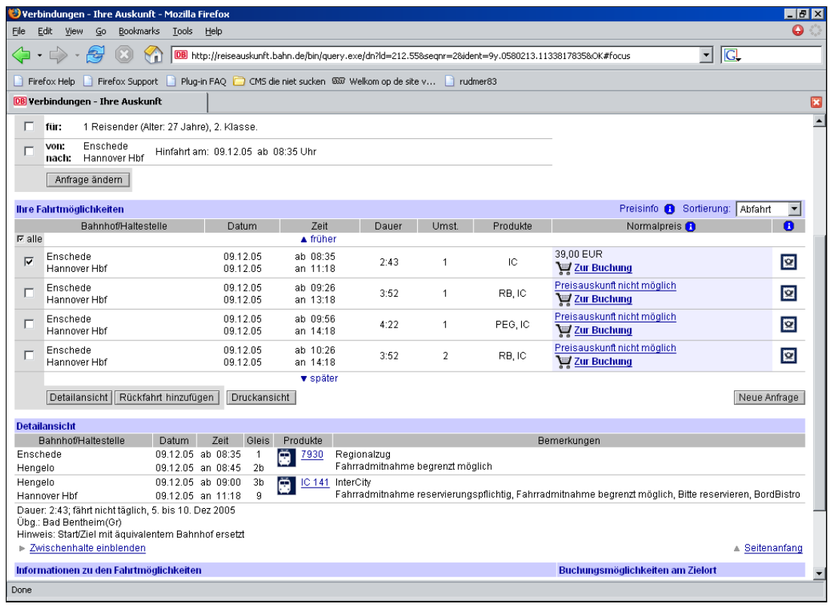
- Increased usage of web applications
- Users open links in new windows.

	1995	1996	Our Study
Link	45.7%	43.4%	43.5%
Back	35.7%	31.7%	14.3%
Submit	-	4.4%	15.3%
New window	0.2%	0.8%	10.5%

Backtracking in Web applications

In contrast to traditional hypertext, the back button is used for undo actions. Often disrupts interaction.

Like in office applications, the interface should provide users with a button to save dynamic content, such as travel plans.



Backtracking and multiple tabs or windows

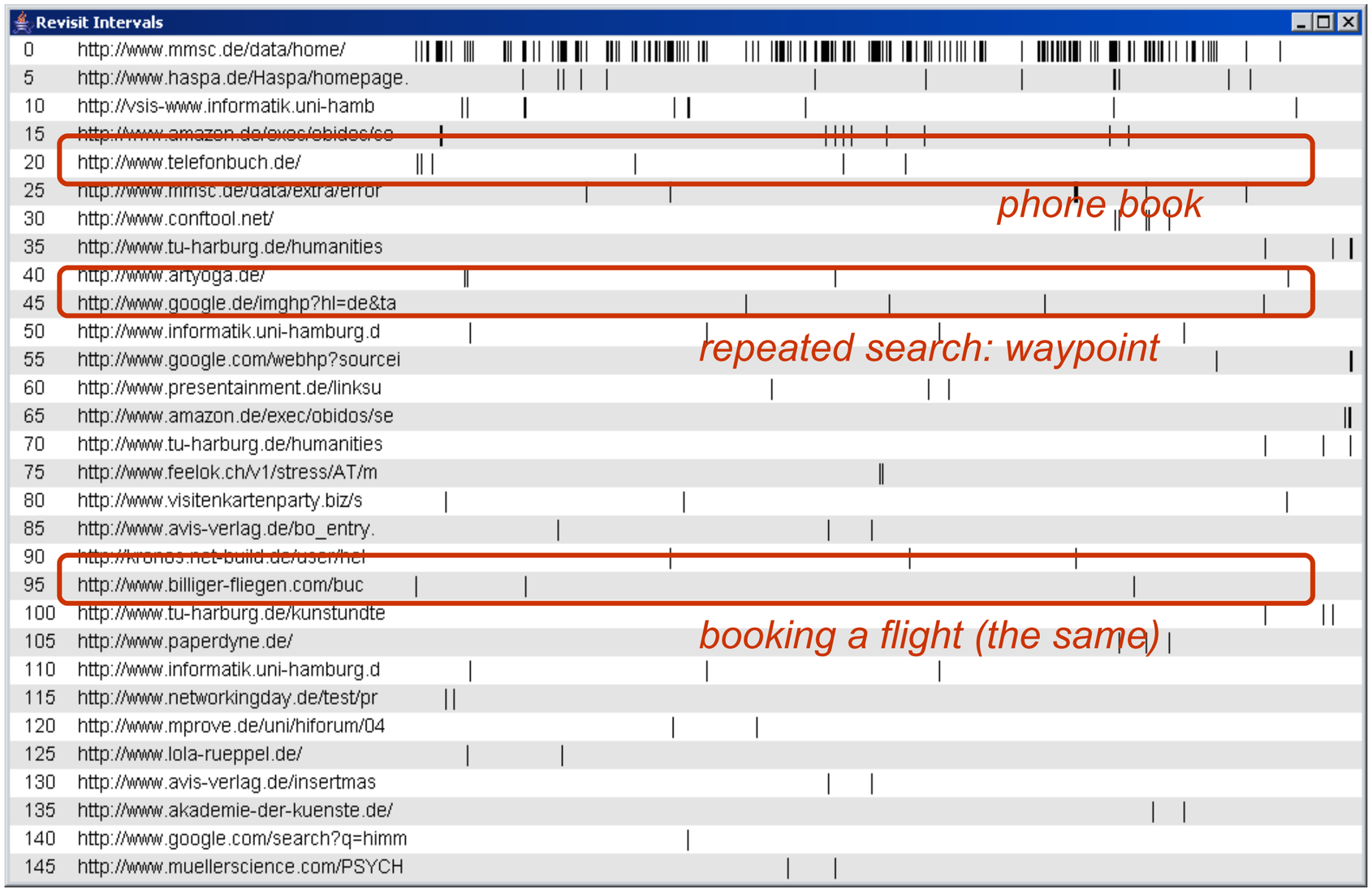
Backtracking is used for returning to an earlier point.

With multiple windows and multiple tabs, the navigation history is split.

Users have to keep track of what they did in which window.

A temporally ordered list would not solve the problem either, as this ignores the fact that users carry out parallel tasks.

There are many less frequently revisited pages

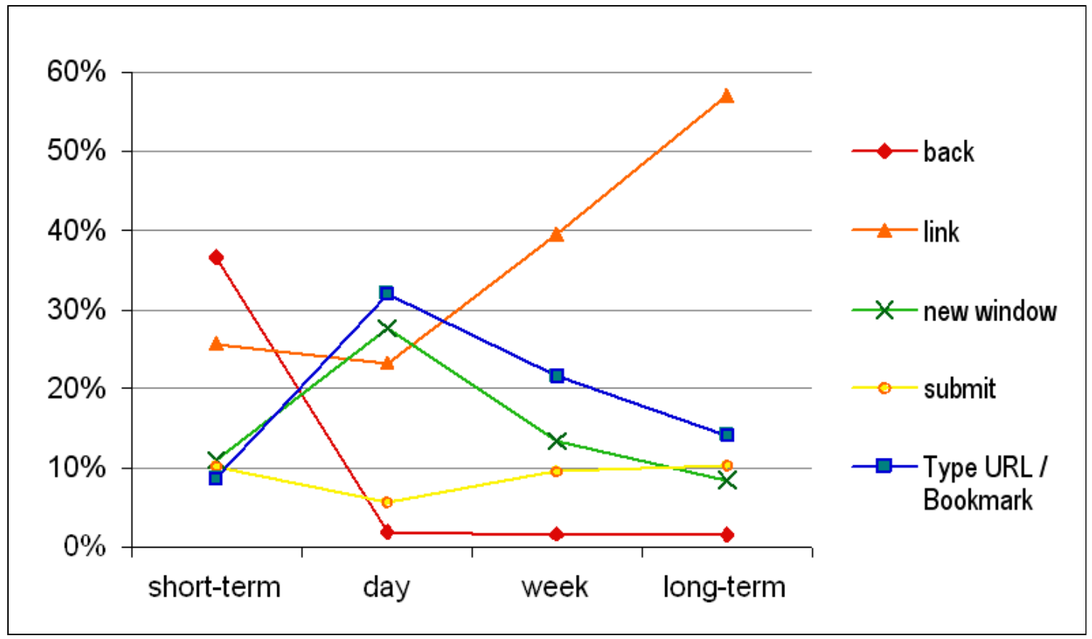


Long-term revisits are hard

Short-term: mainly backtracking activities.

Day and week: url still available for autocompletion

Long-term: users rely on coping strategies.



Prediction Models

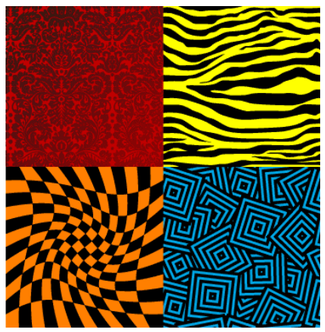
Probabilistic models

Useful to build models that describe the browsing behavior of users

Can generate insight into how we use Web

Provide mechanism for making predictions

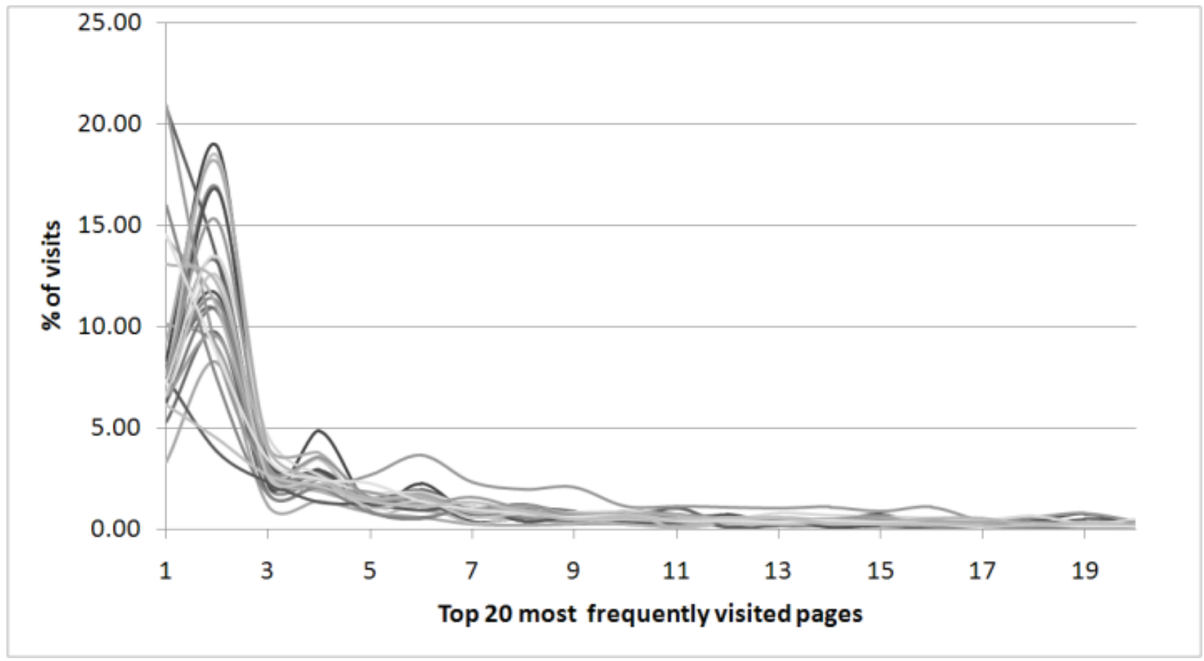
Can help in pre-fetching and personalization



For the most part we are pretty predictable

- We have a small set of pages that we visit frequently.
- Most revisits are to pages that we visited only just before.

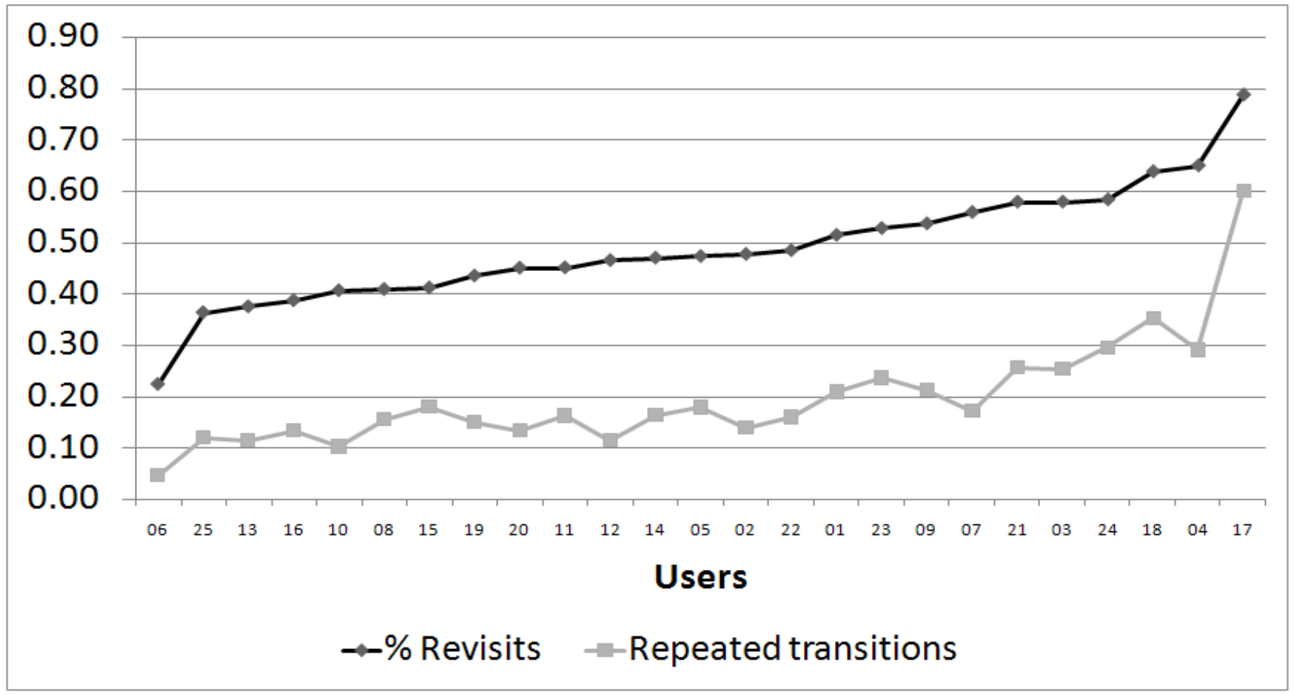
But how useful is that knowledge?



Revisitation and repetitive behavior

Predictive models often assume that users repeat sequences of actions.

That seems a valid assumption



Basic idea

exploit known patterns to predict future actions

Naïve approach

most frequently visited pages, least recently visited pages

Machine learning

association rules, frequent sequences, markov models, support vector machines, ...

We need a method that works on an expanding number of visited pages

Two categories of prediction methods

Ranking methods

Estimate the likelyhood that it will be accessed in the next transaction based on some evidence

e.g. recency or frequency of earlier visits

Propagation methods

Capture repetitiveness in the surfing behavior

Identify groups or sequences of pages

Order-preserving or not

Ranking methods

Least Recently Used (LRU)

Most Frequently Used (MFU)

Polynomial Decay (PD)

Polynomial decay

sum of previous visits to a page, visits from the more distant past get lower weighting

harmonically combines frequency and recency

exponential decay too strong (emphasis on recency)

logarithmic decay too weak (emphasis on frequency)

Popular propagation method: Markov models

General approach is to use a finite-state Markov chain

Each state can be a specific Web page or a category of Web pages

If only interested in the order of visits (and not in time), each new request can be modeled as a transition of states

Issues

- Self-transition
- Time-independence

Markov Models (continued)

For simplicity, consider order-dependent, time-independent finite-state Markov chain with M states

Let s be a sequence of observed states of length L. e.g. s = ABBCAABBCCBBAA with three states A, B and C. st is state at position t (1<=t<=L). In general,

$$P(s) = P(s_1)\prod_{t=2}^{L} P(s_t \mid s_{t-1},\ldots,s_1)$$

Under a first-order Markov assumption, we have

$$P(s) = P(s_1)\prod_{t=2}^{L} P(s_t \mid s_{t-1})$$

This provides a simple generative model to produce sequential data

K-th order Markov chains

First-order Markov model assumes that the next state is based only on the current state

Limitations

- Doesn't consider 'long-term memory'
- We can try to capture more memory with kth-order Markov chain

Limitations

Inordinate amount of training data O(Mk+1)

$$P(s_t \mid s_{t-1},..,s_1) = P(s_t \mid s_{t-1},..,s_{t-k})$$

Transition and association matrices

Transition Matrix

order-preserving, similar to Markov models

simple connectivity

continuous connectivity

a page is connected with all subsequently accessed pages

increasing or decreasing decay parameter

Association Matrix

Assumption: temporal order of transactions in a session not important

Smoothing based on mutual information to reduce influence of very common occurring visits

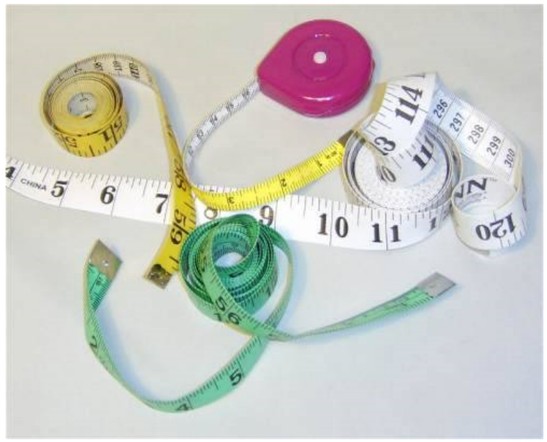
Evaluation of methods using a dataset - measures

Precision at 10 (P@10)

the percentage of revisitations that involved a web page ranked in the top 10 positions

Average Ranking Position Reduction Ratio (RR)

the degree of improvement conveyed by the prediction method in comparison with the actual re-visitation behavior



Performance

Ranking methods

MFU performed worst → recency stronger than frequency

LRU performed better -> obvious

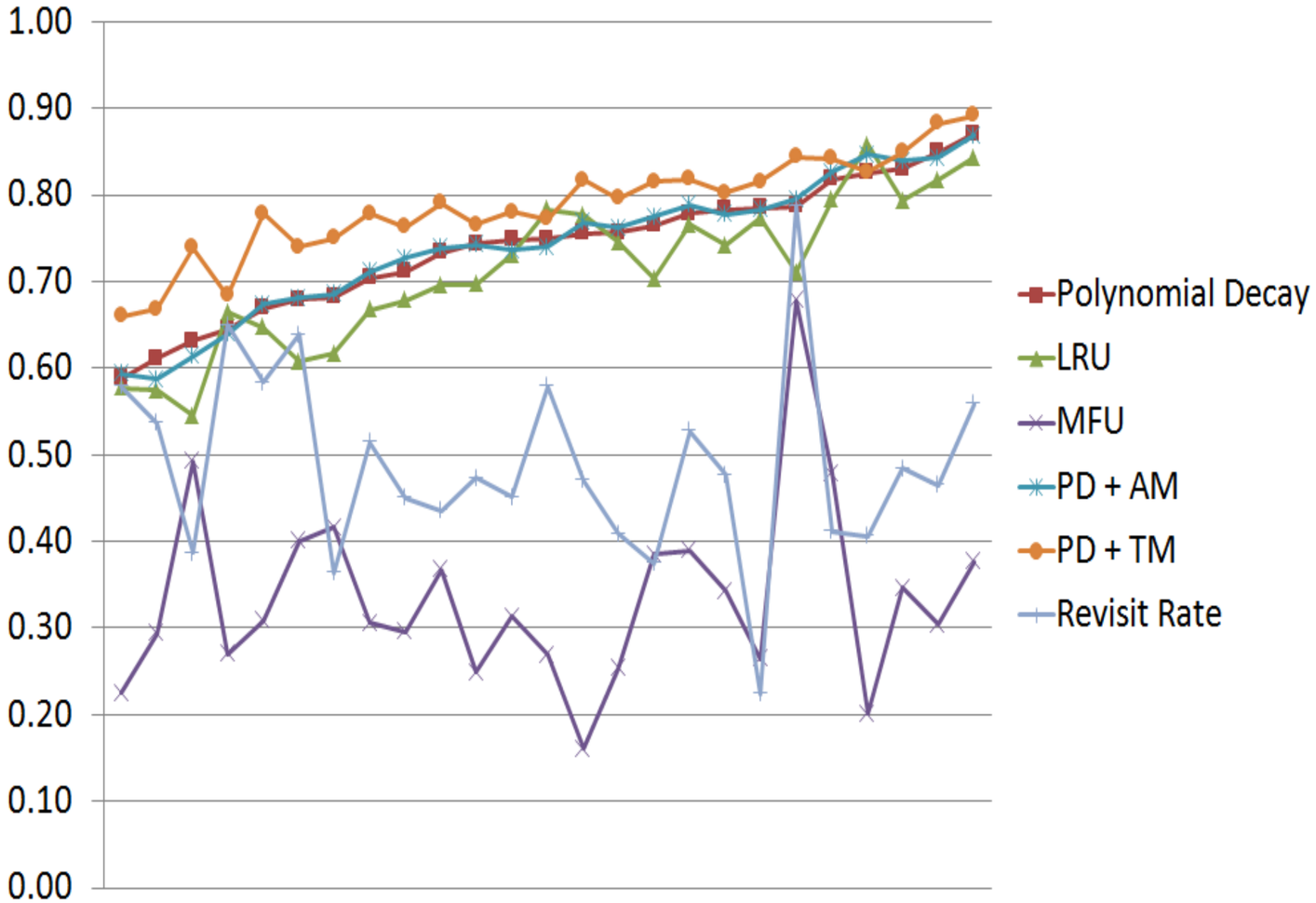
PD exhibits best performance

Combining with propagation methods

PD combined with AM or TM leads to better performance

MFU or LRU combined with AM or TM leads to worse performance

Some users are more predictable than others



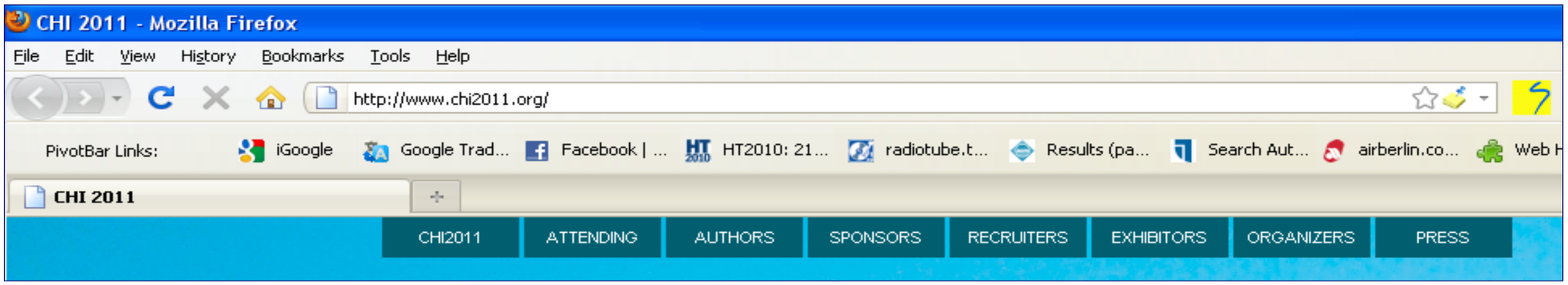
Application: PivotBar

Dynamic toolbar with links and favicons

Changes with each page visit or tab change

Recommendations based on PD+TM

In a pre-test, over 20% of all revisits through the toolbar



Personalization

What is personalization?

Peter Brusilovsky

By adaptive systems we mean all systems which reflect some features of the user in a user model and apply this model to adapt various visible aspects of the system to the user.

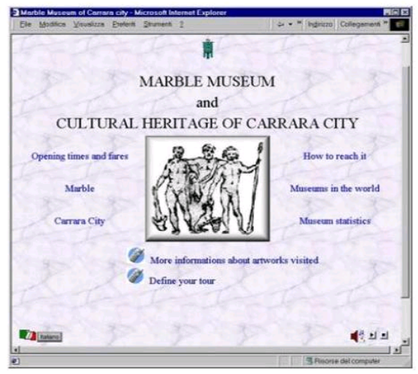
What can be personalized?

Basically, the sky is the limit. Anything that you can think of and that sounds reasonable.

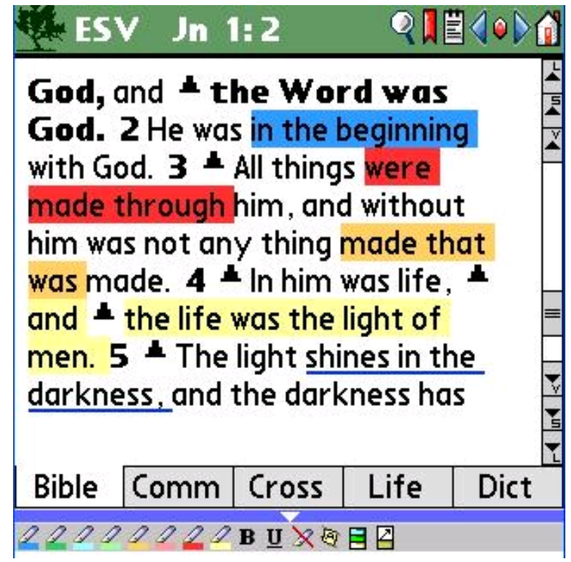
But typically two basic types are distinguished:

Adaptive Presentation Techniques

Adaptive Navigation Support



Adaptive presentation



To make things clearer, more focused or just nicer

Text adaptation

To match the users' interests and knowledge

Hiding, adding, highlighting, annotating, …

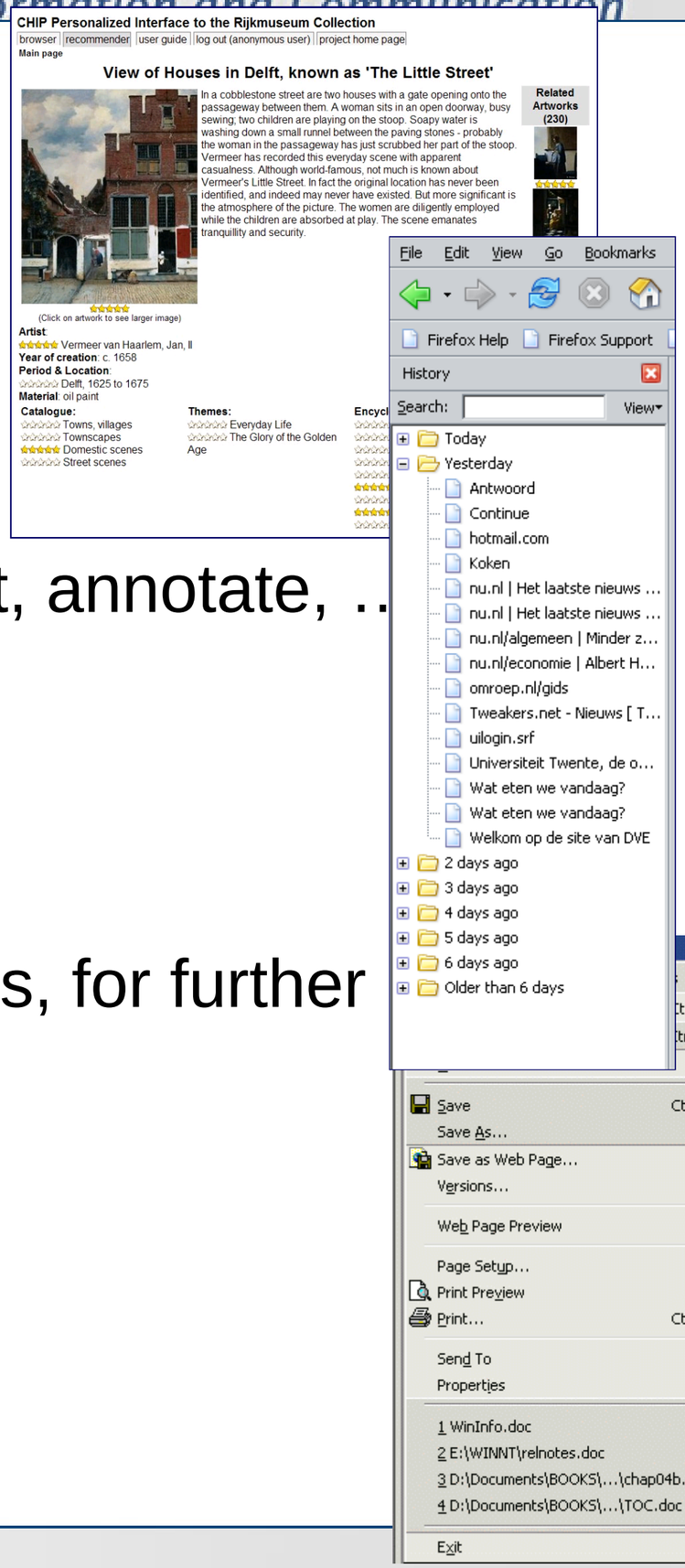
Popular technique: canned text

Adaptation of modality

Text to voice, mobile devices, reduction of images, …

Personalized presentation

Style sheet, contents of choice, background image, alternative designs, skins



Adaptive navigation

To make things easier to find or to refind

Personalized menus: reorder, hide, add, highlight, annotate, …

Personalized search results

Recommendations

Direct guidance

Associative links: to semantically related concepts, for further information (dictionary, encyclopedia, …)

Navigation history

Annotate which links you already followed

Construct trails

History lists

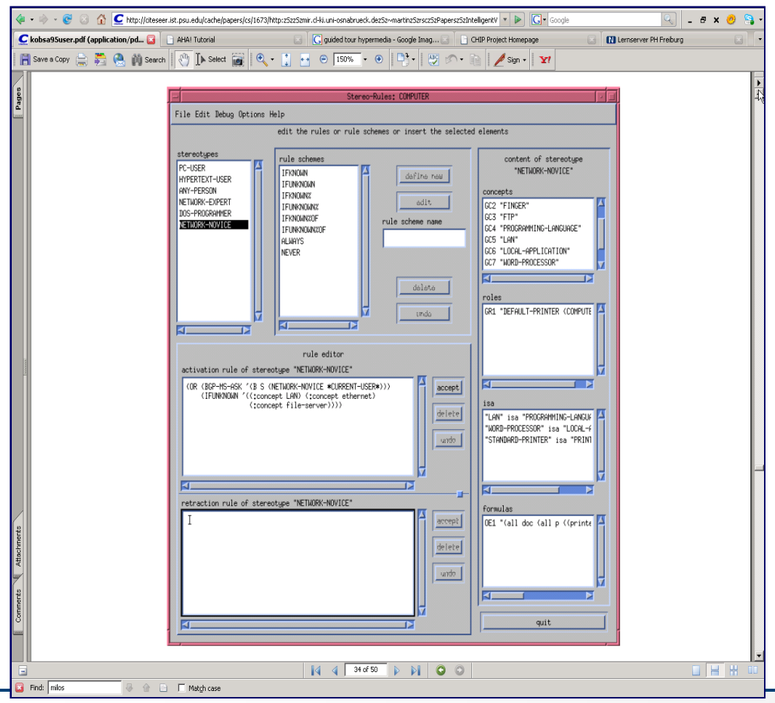
Evolution of user models

Early days: explicit user profiles, hand-crafted and typically based on a mentalistic paradigm

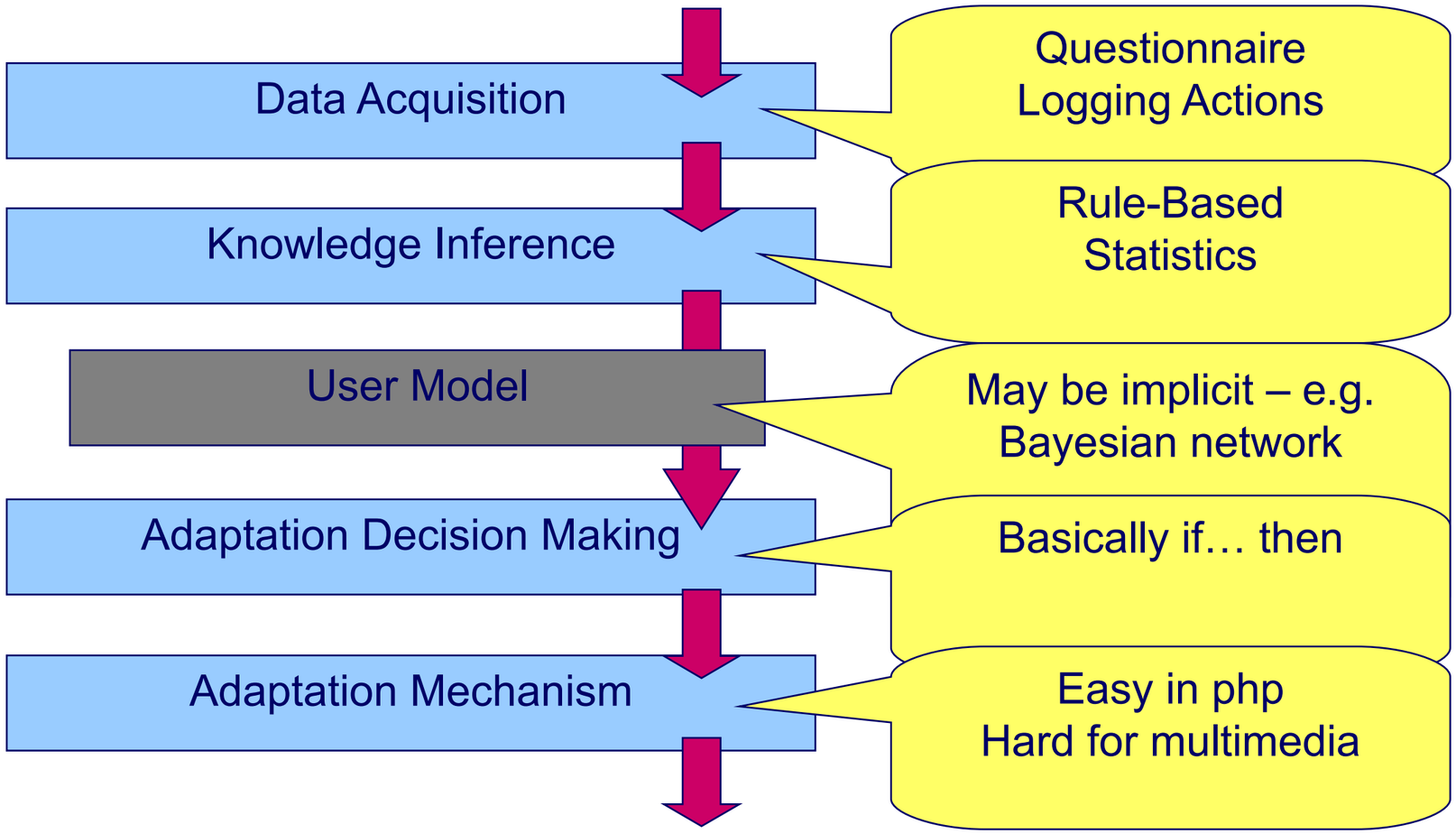
1990s: statistical methods, data mining, Web usage mining.

Pragmatic, light-weight models

Web 2.0 Era: collaborative filtering, tagging, blogging, rating, grouping



The personalization process



Web History Repository

Do you want your Web browser to be smarter?

So do we. And it is easy for you to help.

Submit your anonymized Web history to the Web History Repository.

With your data, researchers can gain new insights; programmers can create new tools and plugins for you.

http://webhistoryproject.blogspot.com

Spread the word via Facebook and Twitter.

Thanks!