Web Dynamics

Part 7 – Human Behaviour on the Web

7.1 Recommendation

7.2 Personalized Search

High-Level View of Recommendation

Input: Collected data on ***behavior of users***

- Items (books, dvds, cds,…) purchased
- Items (books, movies, hotels, …) rated
- Web sites browsed or bookmarked
- Searches and clicked search results
- Sequence of activities (browsing, searching, …)
- Mails, Documents read and written
- Profile in social networks (contacts)

$\Rightarrow$ build extensive ***user models***

High-Level View of Recommendation

Output: Items of ***potential interest*** to user

- Items (books, movies, hotels,…) to purchase/view/visit/…
- Web sites to visit
- Improved search results
- Potential query expansions/refinements
- People to meet in social networks

Three orthogonal approaches

User-centric approach („nearest neighbors“):

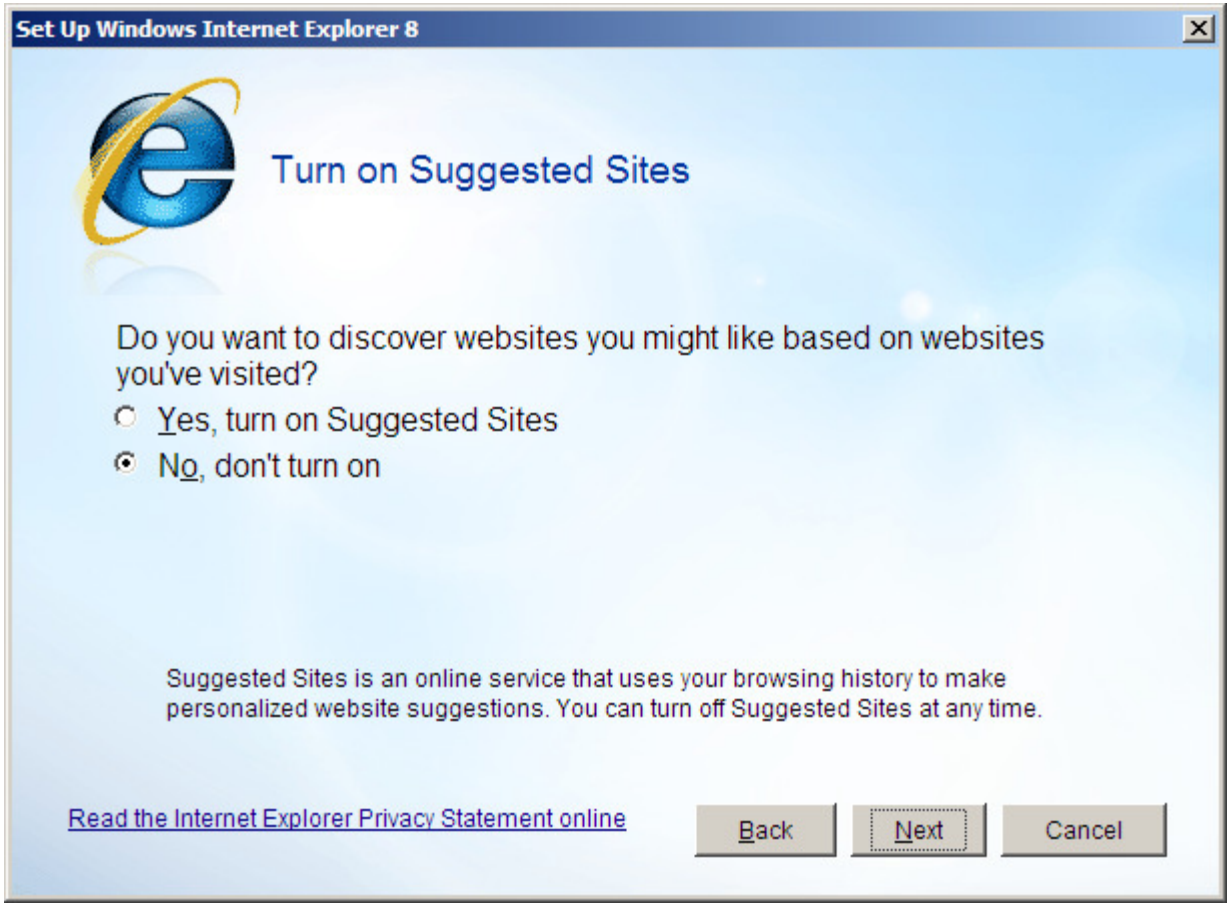
User A likes/buys/visits item X
(model of) user B similar to (model of) user A
} user B may like item X as well

Item-centric approach:

User A likes/buys/visits item X
Item X similar to item Y
} user A may like item Y as well

[***Static approach***: Many people buy X]

Example 1: Web site suggestion



Example 1: Web site suggestion

⇒ item-centric approach, (seemingly) no user model used

Example 2: Product Recommendations

Most Popular in Desktop Computers

Apple iMac MB417LL/A 20...
$1,194.00

HP Pavilion A6700F Desktop PC
~~$731.00~~ **$429.79**

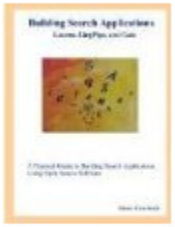
Datenbanksysteme: Eine Einführung
von Alfons Kemper (April 22, 2009)
Auf Lager.

Preis: EUR 39,80
62 Angebote ab EUR 39,80

Gehört mir Kein Interesse x|☆☆☆☆☆ Diesen Artikel bewerten
Diesen Artikel haben wir empfohlen, weil Sie **Übungsbuch Datenbanksysteme** gekauft haben.

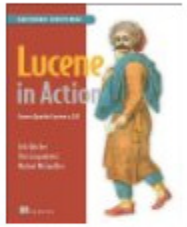
Mehr zu entdecken

Sie haben sich angesehen:

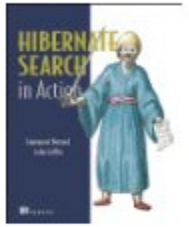


Building Search Applications: Lucene... Taschenbuch von Manu Konchady
EUR 31,89

Ihnen könnten diese Artikel gefallen:



Lucene in Action Taschenbuch von Erik Hatcher, Otis...
EUR 30,95



Hibernate Search in Action Tasche Bernard...
EUR 33,95

› Verwandte Artikel entdecken

⇒ static and item-centric approach

Example 2: Product Recommendation

amazon.com

More to Explore

As someone who's browsed lenses, you may be interested in checking out bestselling digital SLR lenses from Amazon.com.

Canon EF-S 55-250mm f/4.0-5.6 IS Telephoto Zoom Lens for Canon Digital SLR Cameras

Canon EF 50mm f/1.8 II Camera Lens

Nikon 50mm f/1.8D AF Nikkor Lens for Nikon Digital SLR

Nikon 55-200mm f/4-5.6G ED IF AF-S DX VR [Vibration

Canon EF 75-300mm f/4-5.6 III Telephoto Zoom Lens for Canon SLR Cameras

Other products by Canon | See collection

4.1 out of 5 stars See all reviews (140 customer reviews) | More about this product

List Price: ~~$199.99~~

Price: **$155.00** & this item ships for **FREE with Super Saver Shipping**. Details

You Save: $44.99 (22%)

Special Offers Available

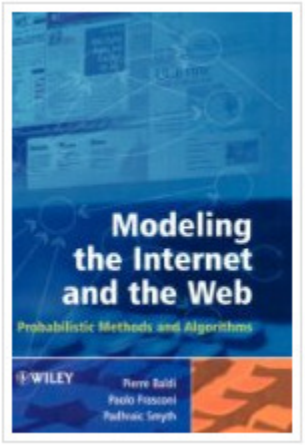
In Stock.

We are not able to ship this item to your default shipping address.

Ships from and sold by **Amazon.com**. Gift-wrap available.

18 new **19 used** from $125.00 **8 refurbished** from $129.90

Example 3: Book Recommendations



Modeling the Internet and the Web: Probabilistic Methods and Algo

by Pierre Baldi

Members	Reviews	Popularity	Ave
10	None	439,588	

▼ Book information

Modeling the Internet and the Web: Probabilistic Methods and Algorithms
by Pierre Baldi
Wiley (2003), Hardcover, 285 pages

LibraryThing recommendations

1. Web Metrics: Proven Methods for Measuring Web Site Success by Jim Sterne
2. Differentiated services for the Internet by Kalevi Kilkki
3. Internet Measurement: Infrastructure, Traffic and Applications by Mark Crovella
4. Designing Campus Networks by Terri Quinn-Andry
5. True Names: And the Opening of the Cyberspace Frontier by Vernor Vinge
6. Me++: The Cyborg Self and the Networked City by Wil
7. What Just Happened: A Chronicle from the Informatior Gleick
8. The Digital Sublime: Myth, Power, and Cyberspace by
9. 24 Hours in Cyberspace: Painting on the Walls of the I Photographed on One Day by 150 of the World's Lead
10. Crypto Anarchy, Cyberstates, and Pirate Utopias by Pe

LibraryThing Recommendations

304 recommendations — page [1] | 2 | 3 | 4

1. **Machine Learning by Thomas Mitchell**
   169 copies. 1 reviews. Average rating 4.08.
   No thanks! | Why? (close why) Recommendation based on:
   Artificial Intelligence: A Modern Approach by Stuart J. Russell
   Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management by Ian H. Witten
   An Introduction to Support Vector Machines and Other Kernel-based Learning Methods by Nello Cristianini
   All of Statistics: A Concise Course in Statistical Inference (Springer Texts in Statistics) by Larry Wasserman

Towards user-centric recommendations

Assume n users U, m items I.

Model ***user-item relation*** as n x m – matrix V:

- $V=\{0,1\}^{nxm}$: binary ***purchase matrix***
- $V=[min,max]^{nxm}$: quantified ***preference matrix***

Both are very sparse!

(Librarything: 1,000,000 users, 52 mio books,
less than 200 books for most users
$\Rightarrow$0,0004% non-zero entries)

„semantics“: v_{ij} seen as „vote“ of user i for item j

Recommendation Problem

Inputs:

- Set of votes of user u with items I_u
- Set of votes of other users

Goal: predict votes of u for items in $I \backslash I_u$
(to identify the items with highest votes)

$\Rightarrow$ yields scalability problem ($|I|$ is large!)

Vote Prediction

Initial vote calibration (to remove bias):

$$v_i = \frac{1}{|I_i|}\sum_{j \in I_i} v_{ij} \qquad\qquad v_{ij}^* = v_{ij} - v_i$$

Predict vote of user u for item j as weighted average over the votes of all other users:

$$\hat{v}_{uj} = v_u + \frac{1}{C}\sum_{i=1}^{n} w_{ui} \cdot v_{ij}^* \qquad\qquad C = \sum_{i=1}^{n} |w_{ui}|$$

similarity of users u and i

Estimating User-User Similarity

- Correlation-Based similarity:

$$w_{ai} = \frac{1}{C_2} \sum_{j \in I_a \cap I_i} (v_{aj} - v_a)(v_{ij} - v_i)$$

$$C_2 = \left(\sum_{j \in I_a \cap I_i} (v_{aj} - v_a)^2 \sum_{j \in I_a \cap I_i} (v_{ij} - v_i)^2 \right)^{1/2}$$

Unreliable results if overlap between users is small

- Vector similarity (cosine):

$$w_{ai} = \sum_{j \in I} \frac{v_{aj}}{\sqrt{\sum_{k \in I_a} v_{ak}^2}} \frac{v_{ij}}{\sqrt{\sum_{k \in I_i} v_{ik}^2}}$$

Remaining problem: high dimensionality (number of users and items)

Reducing Dimensionality: SVD

Replace V by ***rank-k approximation*** of V using SVD:

$$V = A \times S \times B^T$$

A: **user-concept** similarity matrix (n×r)

S: diagonal matrix of **singular values** (with r nonzero entries, where r=rank(V)), corresponding to ***topics***

B^T: **concept-item** similarity (r×m)

Additionally ***restrict to k largest singular values*** to further reduce dimensionality

SVD Example

$$V = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0.707 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0.788 \\ 0 & 0.615 \end{pmatrix} \begin{pmatrix} -0.544 & 0 & 0.707 \\ -0.707 & 0 & -0.5 \\ 0.707 & 0 & -0.5 \\ 0 & -0.615 & 0 \\ 0 & 0.788 & 0 \end{pmatrix} \times \begin{pmatrix} 2.414 & 0 & 0 & 0 & 0 \\ 0 & 2.136 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.662 & 0 \\ 0 & 0 & 0 & 0 & 0.414 \end{pmatrix} \times \begin{pmatrix} 0.5 & 0.5 & 0.707 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.369 & 0.657 & 0.657 \\ -0.707 & 0.707 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.929 & 0.261 & 0.261 \\ 0.5 & 0.5 & -0.707 & 0 & 0 & 0 \end{pmatrix}$$

A S B^T

SVD Example

$$V = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\approx \begin{pmatrix} 0.707 & 0 \\ 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0.788 \\ 0 & 0.615 \end{pmatrix} \times \begin{pmatrix} 2.414 & 0 \\ 0 & 2.136 \end{pmatrix} \times \begin{pmatrix} 0.5 & 0.5 & 0.707 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.369 & 0.657 & 0.657 \end{pmatrix} = \begin{pmatrix} 0.854 & 0.854 & 1.207 & 0 & 0 & 0 \\ 0.604 & 0.604 & 0.854 & 0 & 0 & 0 \\ 0.604 & 0.604 & 0.854 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.621 & 1.106 & 1.106 \\ 0 & 0 & 0 & 0.485 & 0.864 & 0.864 \end{pmatrix}$$

A S B^T

Recommendations with SVD

- Predict votes on A, not on V
  $\Rightarrow$ compute estimate v'_{uj} for each topic j
- Extend the vote estimate from topics to items

$$v_{ui} = \sum_{j=1}^{k} \left(v'_{uj} \cdot S_{jj} \cdot B_{ji} \right)$$

New issue: Maintaining the SVD when data changes

SVD generates implicit clustering of items

Reducing Dimensionality: Clustering

- Reduce number of users by precomputing ***K clusters of similar users***
- Represent each cluster P by its ***centroid*** c(P):

$$c(P)_i = \frac{1}{|P|}\sum_{u \in P} v_{ui}$$

- For prediction:
  - Assign user to one of the clusters
  - Compute „nearest neighbor“-prediction for clusters instead of users
- **Potential problem**:
  users may belong to multiple clusters

User-Centric is Expensive

- User actions are highly dynamic
  - difficult to precompute and maintain similarities
  - best recommendations based on items just bought
- One recommendation takes time O(n+m):
  - needs to scan all users and their items
  - most users have ≤C1 items
  - few users (≤C2) have >C1 items
  - cost bounded by $(n-C2)\cdot C1 + C2\cdot m = O(n+m)$
  - n,m large
- Recommendations need to be computed in real time (≤200ms)

Item-centric Recommendations

Observation:

Relationships of items (i.e., correlation in purchases) a lot less dynamic than relationships of users

- information from yesterday still reasonably accurate today
- not recommending new items tolerable

Predict vote of user u for item j as weighted average over the votes of user u for other items:

$$\hat{v}_{uj} = v_u + \frac{1}{C}\sum_{i=1}^{m} w_{ji} \cdot v^*_{ui} \qquad C = \sum_{i=1}^{m} | w_{ui} |$$

similarity of items j and i

Requires only limited knowledge about the user

Estimating Item-Item Similarity

using correlation-based or cosine similarity

(similar to user-user similarity)

Example: cosine similarity

$$w_{ji} = \sum_{u \in U} \frac{v_{uj}}{\sqrt{\sum_{k \in U} v_{kj}^2}} \frac{v_{ui}}{\sqrt{\sum_{k \in U} v_{ki}^2}}$$

Computing similarities expensive ($O(m^2n)$), but offline

Computing predictions is cheap ($O(m)$ if only constant number of items considered)

Using Search to Recommend

Assume we can identify ***features*** of items (genre, actors, director, keywords, …)

- Identify ***frequent/characteristic features*** for the user‘s items
- Submit ***search*** for those features and recommend the results

Problems:

- Does not scale well for many owned items
- Does not provide good recommendations

Probabilistic Models for Recommendation

Consider ***joint probability distribution*** for m-dimensional set of items (binary preferences):

$P[v_1 \ldots v_m]$: probability that random user has vote vector $(v_1, \ldots v_m)$

Predict unknown value v_{ui} as $P[v_i = 1 | v_j = 1 \text{ for } j \in I_u]$

Impossible to maintain explicitly (2^m parameters!)

$\Rightarrow$approximate through ***finite mixture***:

$$P[v_1 \ldots v_m] \approx \sum_{k=1}^{K} P[v_1 \ldots v_m \mid c = k] \cdot P[c = k]$$

assume independence within each component:

$$P[v_1 \ldots v_m \mid c = k] = \prod_{j=1}^{m} P[v_j \mid c = k]$$

Evaluating Recommender Systems

Goal:

Out of several recommendation algorithms, determine which gives best recommendations.

Required components of such a ***benchmark***:

- set of (user,item,rating) tuples for ***training*** (known to the algorithm in advance)
- set of (user,item,rating) tuples for ***testing*** (where the algorithm needs to predict *rating*)
  - Can be offline (part of the data) or live user experiment
- metrics for ***quantifying result quality***

Properties of Data Sets for Evaluation

- can be ***synthetic*** vs. ***real-life***
- features of the application domain
  - ***novelty*** vs. ***quality*** focus of recommendations
  - ***cost/benefit ratio*** of true/false positive/negatives
  - ***granularity*** of true user preferences (vs. ratings)
- inherent features of the data set (and ratings)
  - ***Implicit*** or ***explicit*** ratings
  - ***scale*** & ***dimensions*** of ratings
  - ***history*** of ratings (timestamps) and recommendations
- sample features
  - ***density*** of rating set (overall & for test users)
  - ***size*** of data set

Offline Evaluation vs. User Experiments

- ***Offline evaluation***: ***compare*** predicted votes to actual votes made by the user
  - low effort, can be done automatically
  - can be used to evaluate series of ratings (timestamps)
  - But: limited choice of predictions to evaluate
- ***Live user experiments***: ***ask*** user for opinion or ***observe*** user behavior
  - understand if and why people like (or dislike) recommendations, interfaces, systems

Evaluation metrics

Widely used: measure ***accuracy*** of predictions by measuring the ***error*** of prediction and actual rating

- mean absolute error (MAE)

$$|\overline{E}_{MAE}| = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N}$$

p_i: prediction
r_i: recommendation
N: # recommendations

- Root mean square error (RMSE; emphasises large errors)

$$|\overline{E}_{RMSE}| = \sqrt{\frac{\sum_{i=1}^{N} (p_i - r_i)^2}{N}}$$

- precision/recall, rank accuracy metrics, …

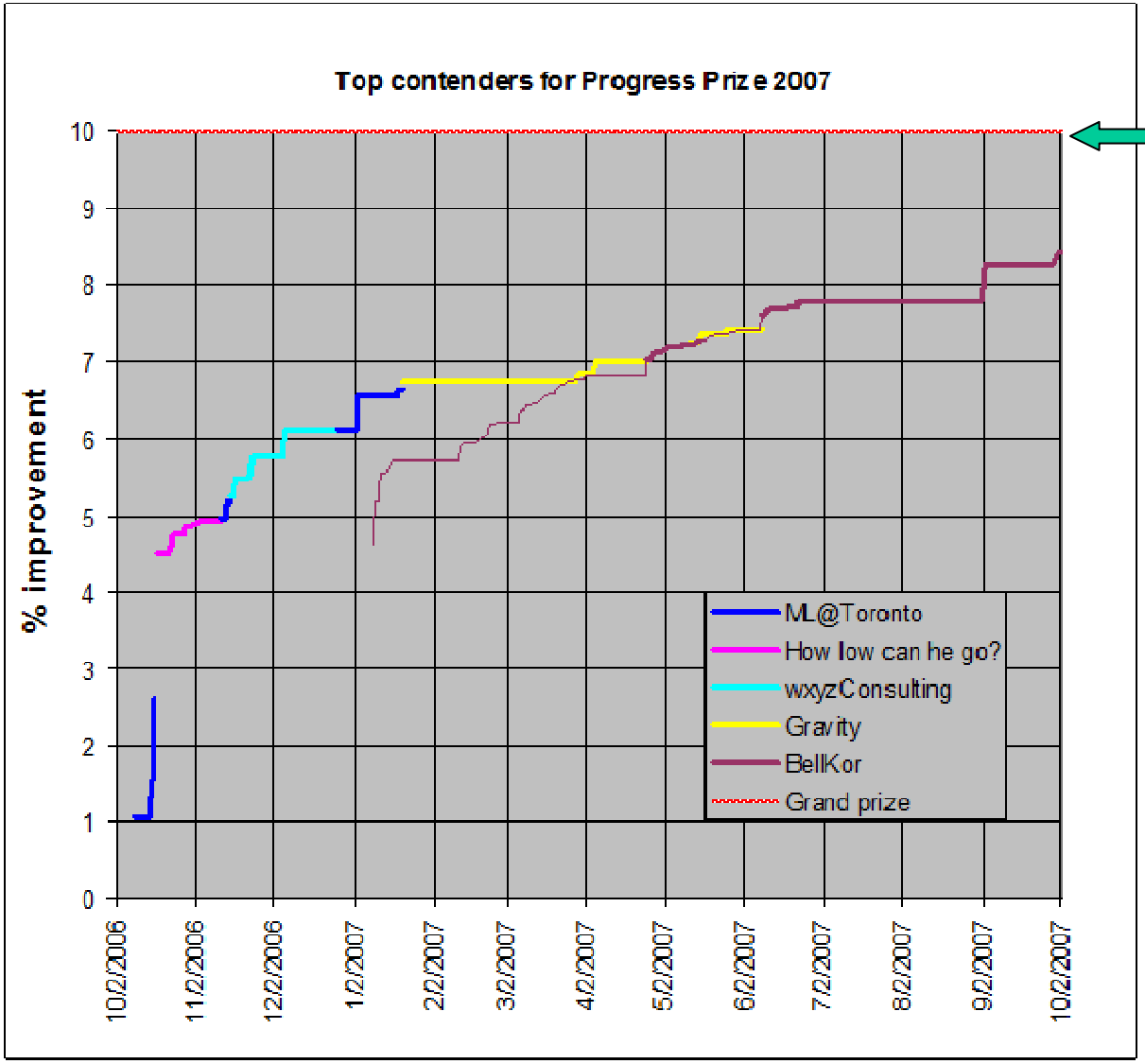
Additional Evaluation Dimensions

- ***Coverage***:
  - Recommendations for how many items
  - How many items are actually recommended
- ***Learning rate***:
  - How fast recommendation quality increases with increased amount of training data
- ***Novelty***:
  - Focus on items unknown to the user, but within its scope (e.g., new movie of favourite director)
- ***Serendepity***:
  - Surprising recommendations (e.g., new movie of new director that fits the user‘s taste)
- ***Confidence***

Benchmarks: Netflix Prize

- http://www.netflixprize.com
- set up by online movie portal
- provides (anonymized) training data (480,000 users, 18,000 movies, 10^6 ratings on a 1..5 scale)
- Goal: improve over portal's own recommender (RMSE: 0.9514)
- High reward to make the benchmark attractive: 1,000,000$ for the first 10% improvement in RMSE on test data (1.4 million user-movie pairs), 50,000$ intermediate progress award per year

Netflix: Result Improvements over Time



10% improvement reached on July 26, 2009

Web Dynamics

Part 7 – Human Behaviour on the Web

7.1 Recommendation

7.2 Personalized Search

Goal: Resolve inherent disambiguity of search

Example 1: Search for „IR“ may return

- Ingersoll-Rand Company
- Web pages in Arabic from Iran (*.ir)
- Infrared Light
- Information Retrieval

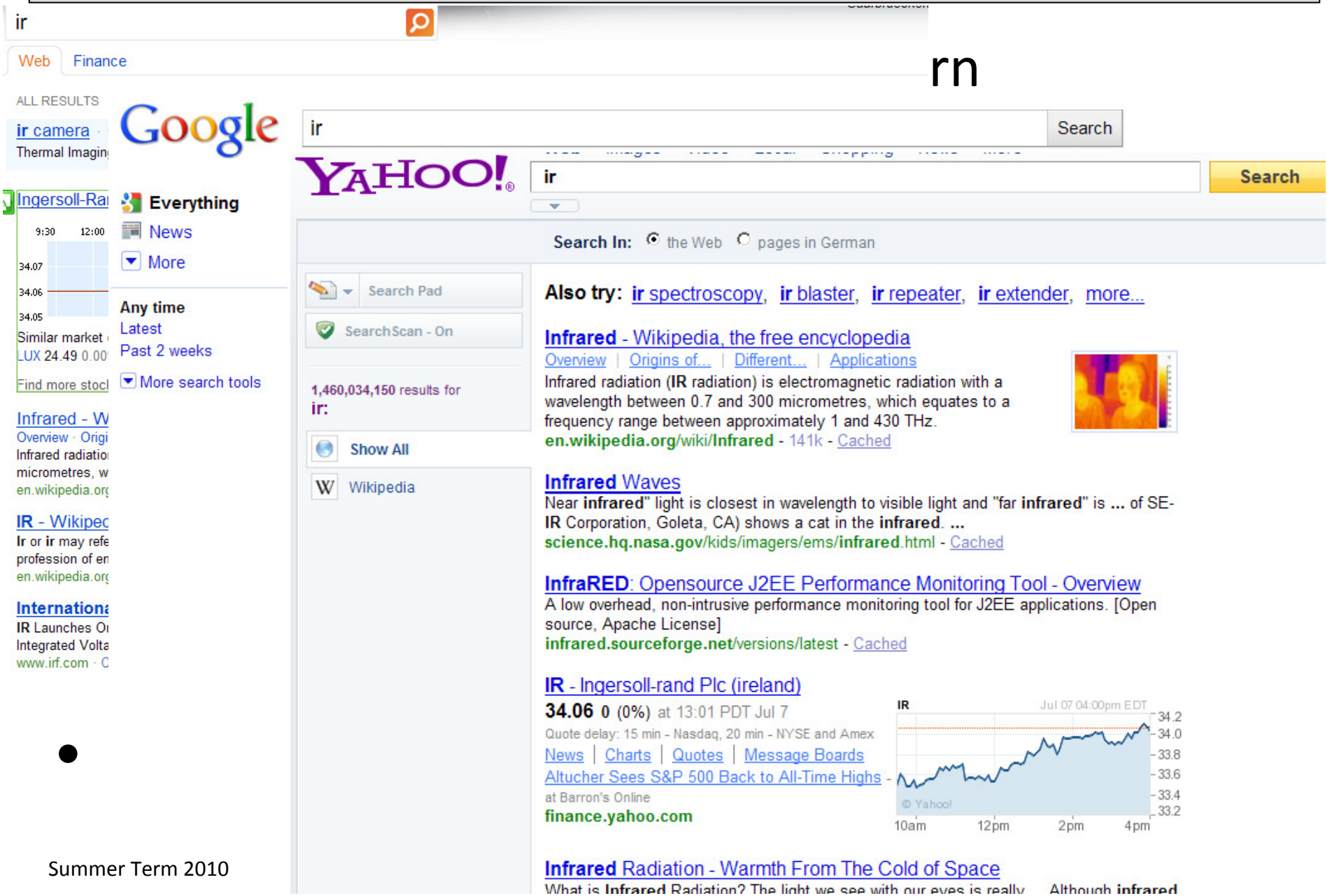
Example 2: Search for „Java“ should return

- Programming tools for a programmer
- Tutorials for a teacher
- FAQ lists for a novice user

global context of the user

Goal: Resolve inherent disambiguity of search

rn



Goal: Resolve inherent disambiguity of search

Example 3: Search for „restaurant“ should return

- places in Geneva while planning for SIGIR 10
- places in Singapore while planning for VLDB 10
- places in Saarbrücken otherwise

Example 4: Search for „Saarbrücken“ should return

- Restaurants (when I‘ve been searching for them)
- Computer shops, dentists, hospitals, …

Search results may depend on current context (that is not constant and may change over time)

Dimensions of Personalized Search

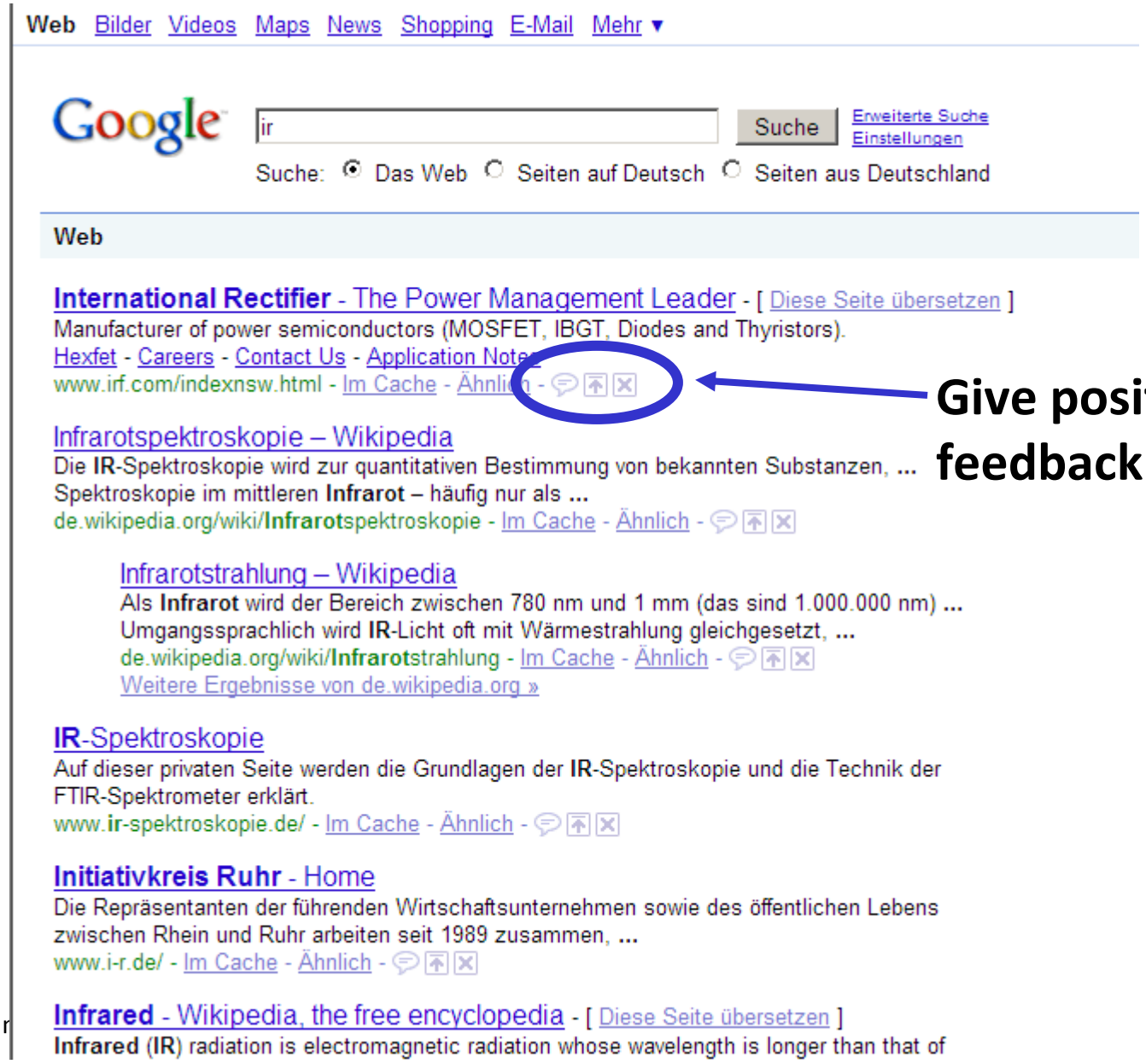
- Different ***kinds of user contexts***:
  - global: background of the user, long-term profile
  - session: set of queries following similar needs
  - query: use last query & actions

  each only for searches, for all browser actions, or for (more/all) actions
- Different ***places*** to collect & use context info:
  - Service provider vs. Web server vs. local client
- Different ***actions*** to use context info:
  - modify query vs. rerank results

Simple Personalization: Relevance Feedback

- collect ***feedback from user*** for query results
  - explicit feedback (buttons in the interface)
  - implicit feedback (clicks of the user)
- generate ***improved query***
  - add new terms
  - drop some old terms
  - change weights of terms

Example: Simple Feedback on iGoogle



Give positive or negative feedback for results

Example: Simple Feedback on iGoogle

Feedback in the current Google interface

iGoogle: Collaborative Feedback

Implicit Feedback from Clicks

General rules to collect implicit feedback:

- ***Clicked results*** are relevant for the query
  - unless the user left that page immediately
- ***Non-clicked*** results don‘t really help
  - User may immediately rate them as nonrelevant (from the snippet)
  - User may already know the result (which may be relevant or nonrelevant)
  - User may not have looked at the result (was satisfied by other results)

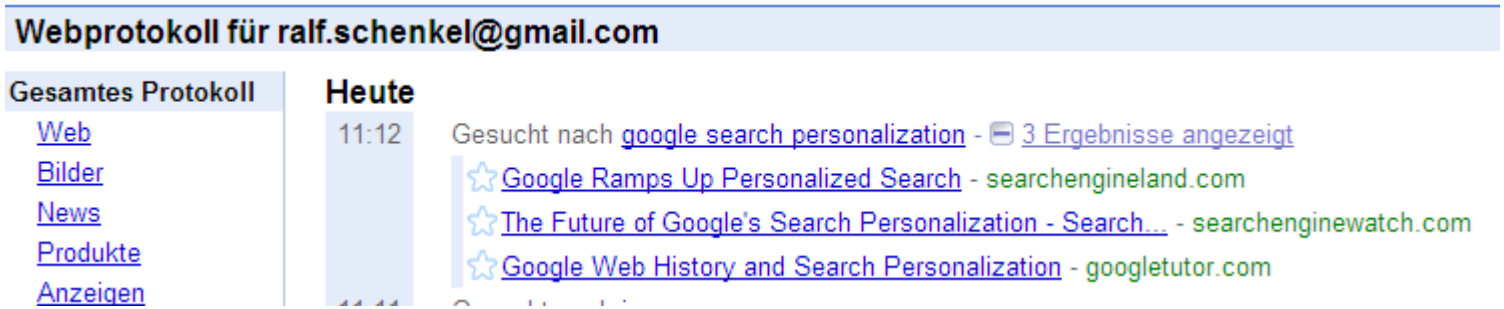
Advanced Implicit Feedback

Modify browser to collect ***behavior data***:

- dwelling time on a page
- scrolling
- mouse movements
- mouse clicks
- followed links

$\Rightarrow$ Yields better estimate of “relevance”

iGoogle: Logging Searches and Clicks



Context-based Search on Google

Google **Personalisierte Suche: ir**

Wenn möglich, passt Google Ihre Suchergebnisse anhand des Standortes und/oder Ihrer aktuellen Suchvorgänge an. Wenn Sie in Ihrem Google-Konto angemeldet sind, erhalten Sie außerdem basierend auf Ihrem Suchverlauf meist noch relevantere und nützlichere Ergebnisse.

Die folgenden Informationen wurden verwendet, um Ihre Suchergebnisse für **ir** zu verbessern:

Webprotokoll	Ein oder mehrere Elemente in Ihrem Webprotokoll wurden zur Verbesserung der Suchergebnisse verwendet. Webprotokoll verwalten Webprotokoll aus meinem Google-Konto entfernen

Wenn Sie neugierig sind, können Sie sich anschauen, wie eine Suche nach **ir** ohne Anpassungen aussieht.

Der Link 'Weitere Details' auf der Suchergebnisseite kann verwendet werden, um diese Seite für etwa 30 Minuten anzuzeigen. Danach wird diese Seite nicht länger angezeigt.

Markierte Ergebnisse für **ir**

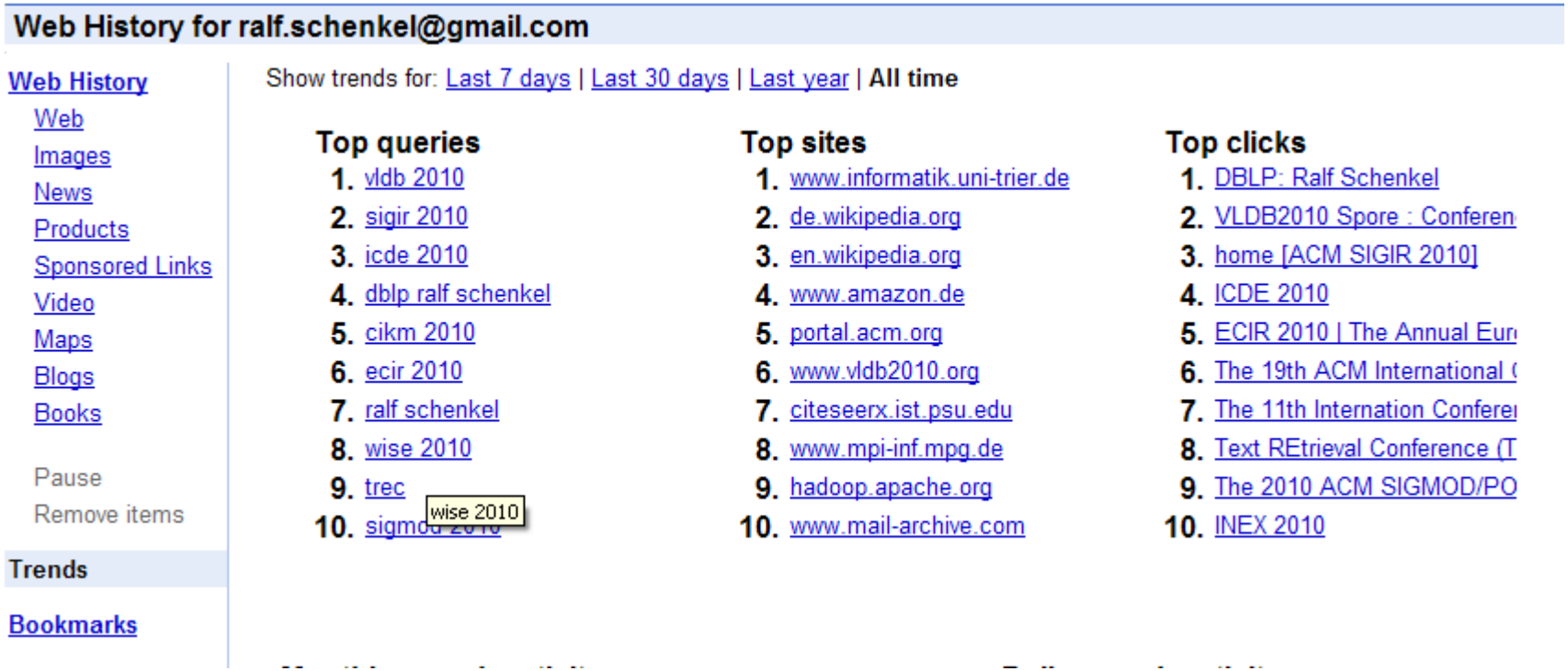
☆ Infrarot-Strahlung (IR) - www.bfs.de/de/uv/ir

Infrarotspektroskopie – Wikipedia ☆
Zu Lage der IR-Absorptionsbanden springen: **IR**-Spektren werden dahingehend interpretiert, dass man aus der Kurve des gemessenen **IR**-Spektrums die ...
Varianten - Messprinzip - IR-Spektroskopie-Techniken - Siehe auch
de.wikipedia.org/wiki/**Infrarot**spektroskopie - Im Cache - Ähnliche

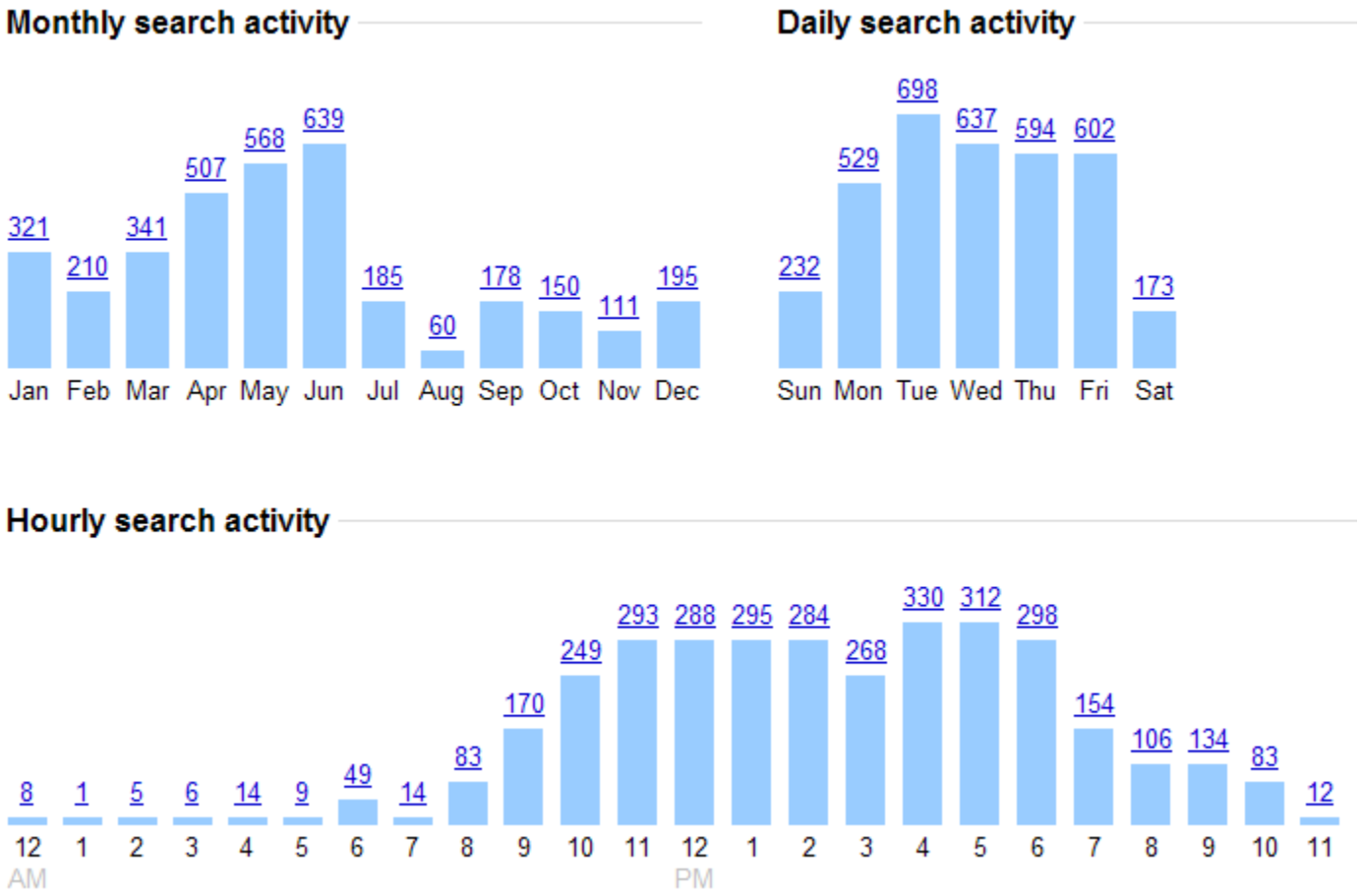
Infrarotstrahlung – Wikipedia ☆
Als Infrarotstrahlung (kurz **IR**-Strahlung, auch Ultrarotstrahlung) bezeichnet man in der Physik elektromagnetische Wellen im Spektralbereich zwischen ...
de.wikipedia.org/wiki/**Infrarot**strahlung - Im Cache - Ähnliche

Weitere Ergebnisse anzeigen von de.wikipedia.org

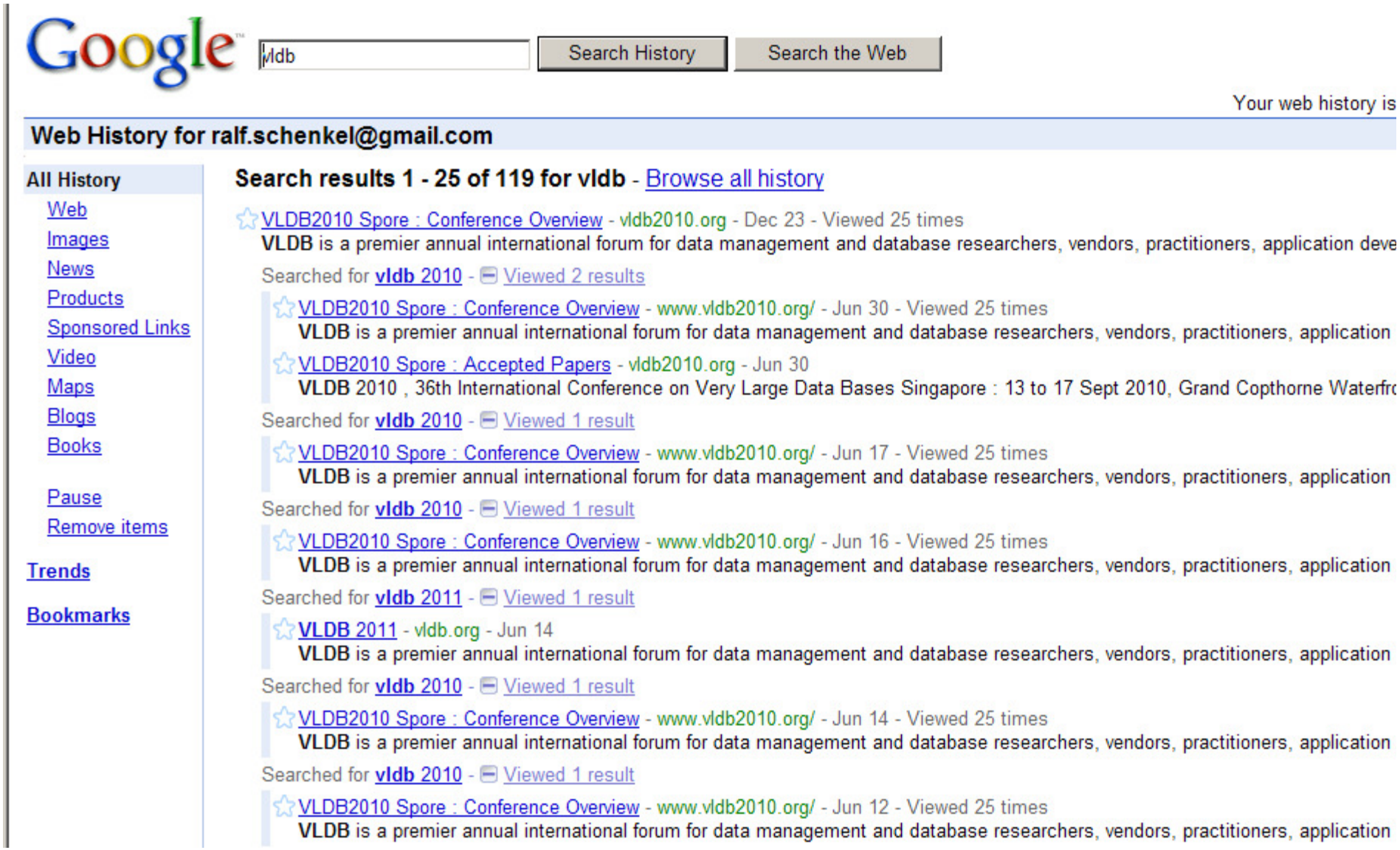
Personal Google Web history



Personal Google Web history



Searching personal histories



Google vldb Search History Search the Web

Your web history is

Web History for ralf.schenkel@gmail.com

All History
Web
Images
News
Products
Sponsored Links
Video
Maps
Blogs
Books

Pause
Remove items

Trends

Bookmarks

Search results 1 - 25 of 119 for vldb - Browse all history

VLDB2010 Spore : Conference Overview - vldb2010.org - Dec 23 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application deve

Searched for **vldb** 2010 - Viewed 2 results

VLDB2010 Spore : Conference Overview - www.vldb2010.org/ - Jun 30 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

VLDB2010 Spore : Accepted Papers - vldb2010.org - Jun 30
VLDB 2010 , 36th International Conference on Very Large Data Bases Singapore : 13 to 17 Sept 2010, Grand Copthorne Waterfr

Searched for **vldb** 2010 - Viewed 1 result

VLDB2010 Spore : Conference Overview - www.vldb2010.org/ - Jun 17 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

Searched for **vldb** 2010 - Viewed 1 result

VLDB2010 Spore : Conference Overview - www.vldb2010.org/ - Jun 16 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

Searched for **vldb** 2011 - Viewed 1 result

VLDB 2011 - vldb.org - Jun 14
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

Searched for **vldb** 2010 - Viewed 1 result

VLDB2010 Spore : Conference Overview - www.vldb2010.org/ - Jun 14 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

Searched for **vldb** 2010 - Viewed 1 result

VLDB2010 Spore : Conference Overview - www.vldb2010.org/ - Jun 12 - Viewed 25 times
VLDB is a premier annual international forum for data management and database researchers, vendors, practitioners, application

Standard RF: Rocchio's Method (1971)

- Goal: Find query that is close to relevant documents
- Compute ***Rocchio weights [1971]*** for each term (also used as weight in query):

$$w(t) = \alpha \cdot q(t) + \beta \frac{r_t}{R} - \gamma \frac{n_t}{N}$$

where

$q(t)$	weight of term t in the query
r_t	number of relevant results with term t
R	number of relevant results
n_t	number of nonrelevant results with term t
N	number of nonrelevant results

- Select n terms with highest weight to expand query

Simple Use of Feedback: Promoting

Idea: Push results with positive feedback up

- ***Locally*** for each user:
  - remember feedback for each user
  - promote results with feedback when query returns (approximately 30% of queries [Dou, WWW07])
- ***Globally*** for all users:
  - collect feedback for (frequent) queries
  - promote results with feedback from „most“ users
  - does not work well for ambigous queries

$\Rightarrow$ pure reranking approach

User Profiles

Goal: Construct ***summary*** of the user's interests

- from the pages she accessed
- from her documents, her mails, … (optional)

General approach:

- For each page p, consider term vector t(p)
- For set of browsed pages B, compute average term vector t(B):

$$t(B) = \frac{1}{|B|} \sum_{p \in B} t(p)$$

Persistent vs. Session Profile

Long-term interests of user may differ from interest in ***current search session***

$\Rightarrow$ maintain two profiles: persistent & session

- **Session profile**:
  - consider pages accessed in the current session only
  - Session boundaries by time or page coherence
- **Persistent profile**:
  - consider all pages ever visited by the user
  - lower weight for older pages (exponential decay)

Profile is mixture of session & persistent profiles

Personalization with User Profiles

Reranking of search result based on profile match:

- compute set of results R for query
- for each result p, measure ***similarity*** of p with profile vector (e.g., cosine)
- rank results in descending order of similarity

Improving Profiles by Collaborative Filtering

Problem:

User profile often sparse (based on few pages)

Approach:

Predict missing term weights analogously to user-centric recommendation

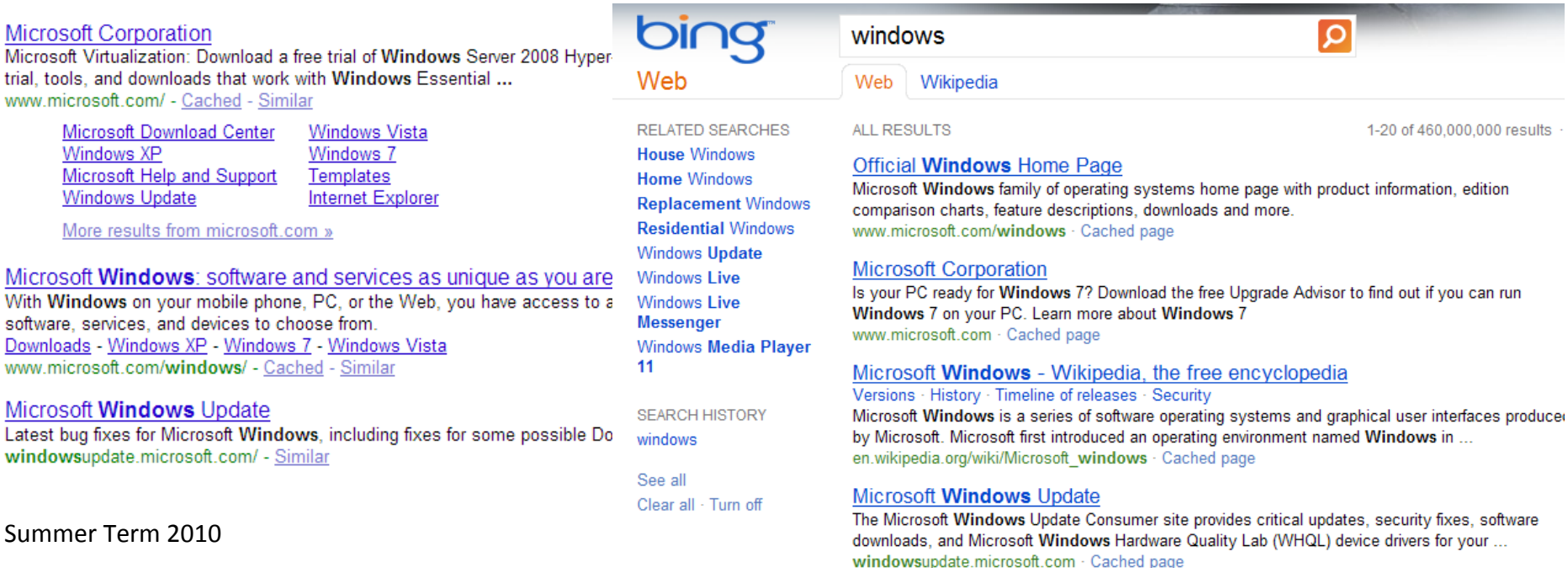
- Find similar users based on similarity of their profiles
- Compute predictions for term weights based on weighted average over neighborhood

Reranking Problem: Similar Results

Reranking cannot work when all results are similar (and nonrelevant to the query)

Example:

- Query: windows (as built into houses)
- Results: only about the operating system



Reranking diversified results

Exploit information about query sequences

Example:

windows $\rightarrow$ house windows $\rightarrow$ vinyl windows
$\rightarrow$ windows xp $\rightarrow$ windows vista

Approach:

To get K results for reranking for query q,

submit top-$K/(k+1)$-queries for k most

frequent/diverse following queries of q in the log

References

- P. Baldi et al: *Modeling the Internet and the Web*, chapter 8
- J.B. Schafer et al.: *E-Commerce recommendation applications*, Journal of Data Mining and Knowledge Discovery 5,115-153, 2001
- B.M. Sarwar et al.: *Analysis of recommender algorithms for e-commerce*, 2nd ACM Conf. On Electronic Commerce, 2000
- G. Linden et al.: *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*, IEEE Internet Computing 7(1), 2003
- R.M. Bell and Y. Koren: *Scalable Collaborative Filtering with Jointly Derived neighborhood Interpolation Weights*, ICDM Conference, 2007
- J. Herlocker et al.: *Evaluating Collaborative Filtering Recommender Systems*, ACM Transactions on Information Systems 22, 2004
- K. Sugiyama et al.: *Adaptive web search based on user profile constructed without any effort from users*, WWW Conference, 2004
- P. Brusilovsky et al. (eds.): *The Adaptive Web*, Lecture Notes in Computer Science 4321, 2007
- Z. Dou et al.: *A Large-scale Evaluation and Analysis of Personalized Search Strategies*, WWW Conference, 2007.
- F. Radinski et al.: *Improving Personalized Web Search using Result Diversification*, SIGIR Conference, 2006.