DAAD Summer School on Current Trends in Distributed Systems (CTDS'2009)
24 - 26 September, Gammarth, Tunisia

Machine Learning in Search

Thomas Hofmann
Engineering Director
Google, Switzerland
thofmann@google.com

Motivation & Overview

Digital revolution

- Digital revolution: **production**, **storage** & **accessibility** of knowledge.
- Digital collections replace libraries, digital content creation, online publication
- Increase in comprehensiveness, freshness, distribution, accessibility, usability, applications

WORLD INTERNET USAGE AND POPULATION STATISTICS

World Regions	Population (2009 Est.)	Internet Users Dec. 31, 2000	Internet Users Latest Data	Penetration (% Population)	Growth 2000-2009	Users % of Table
Africa	991,002,342	4,514,400	**65,903,900**	6.7 %	1,359.9 %	3.9 %
Asia	3,808,070,503	114,304,000	**704,213,930**	18.5 %	516.1 %	42.2 %
Europe	803,850,858	105,096,093	**402,380,474**	50.1 %	282.9 %	24.2 %
Middle East	202,687,005	3,284,800	**47,964,146**	23.7 %	1,360.2 %	2.9 %
North America	340,831,831	108,096,800	**251,735,500**	73.9 %	132.9 %	15.1 %
Latin America/Caribbean	586,662,468	18,068,919	**175,834,439**	30.0 %	873.1 %	10.5 %
Oceania / Australia	34,700,201	7,620,480	**20,838,019**	60.1 %	173.4 %	1.2 %
WORLD TOTAL	6,767,805,208	360,985,492	**1,668,870,408**	**24.7 %**	362.3 %	100.0 %

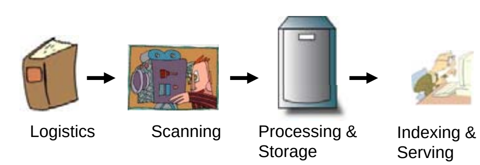
Side note: Google books

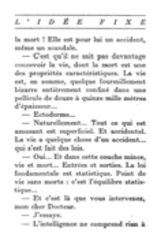
Create a **universal digital library** for the world.

Fictive project plan for digitizing books.

Currently approx. **10 M scanned** (2 trillion words)
40 libraries, 25K partners

Number of books	30,000,000
Years of project	10
Books per year	3,000,000
Books per day	12,000
Pages per book	330
Pages per day	3,960,000
Image size per page	5
TBs a day	20
PBs per year	5
Pbs for project	50
Market cost per book	50
Cost of project at Market rate	1,500,000,000





Search as a principle & problem

"The difficulty seems to be, not so much that we publish unduly in view of the extent and variety of present day interests, but rather that publication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships."

We live in a **search society** – belief that (almost) everything is known, we just have to find the information

We **search for everything** – the right book, movie, car, house, vacation trip, bargain, partner, search engine etc.

NEU.DE
DIE PARTNERBÖRSE

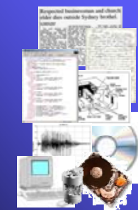
V. Bush. *As we may think*. Atlantic Monthly, 176 (1945), pp.101-108

Machine learning in search

syntax

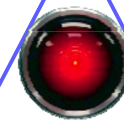
semanti

bits & bytes



001001001010
100011000101

un-interpreted data - text, images, etc.



machine learning

information & knowledge

interpreted data – meaning, interest, intention, know-ledge, information need

interpretation

unsupervised learning & data mining: discover hidden regularities, generate semantically meaningful representations, predictive modeling, statistics

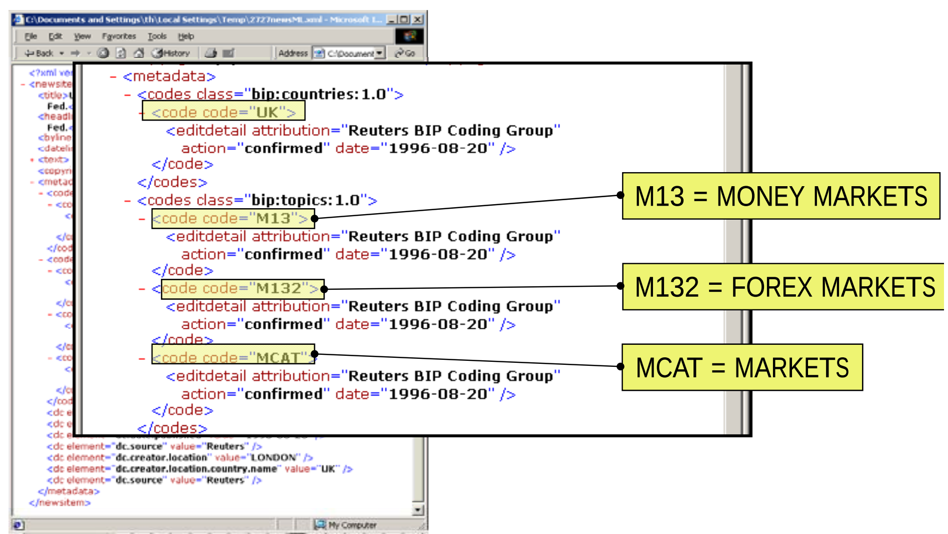
generalization

supervised learning: generalize from given examples, classification & recognition, emulate human experts

1. Text Categorization

Document Annotations

- Categories as metadata: example, Reuters news stories



Text categorization & taxonomies

Business Taxonomies

Document Classification

Medical Terminology

Digital Libraries

Patent Classification

Email folders

Web Directories

Help Desks CRM

Semantic Web

Tasks:

- Assign documents to one of more pre-defined categories
- Route messages to an appropriate expert, employee, or department
- Automatically organize content into folders

Types of texts:

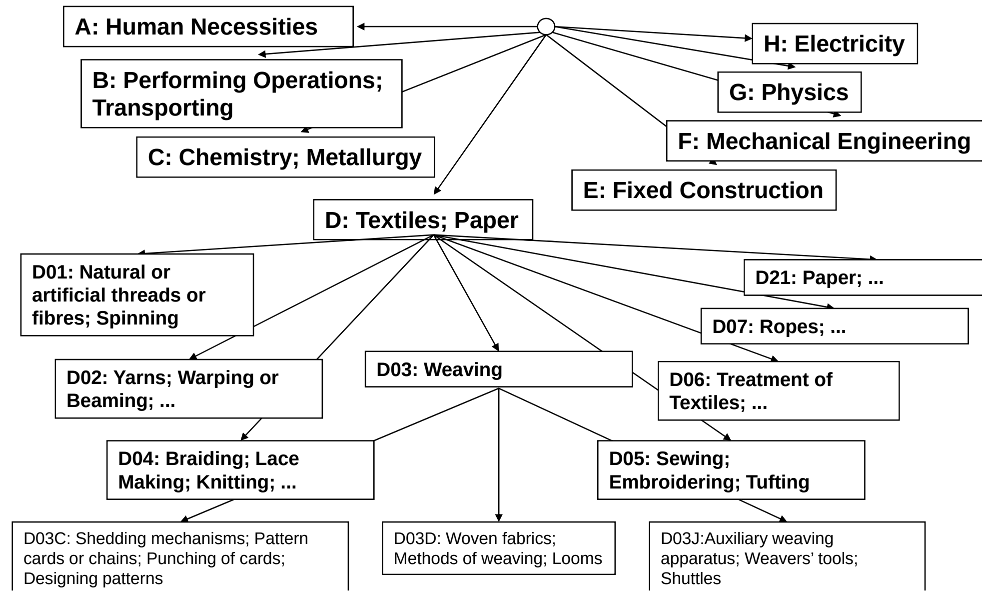
- text documents
- web pages, web sites
- messages, emails, SMS, chat transcripts
- passages & paragraphs, sentences

Types of categories

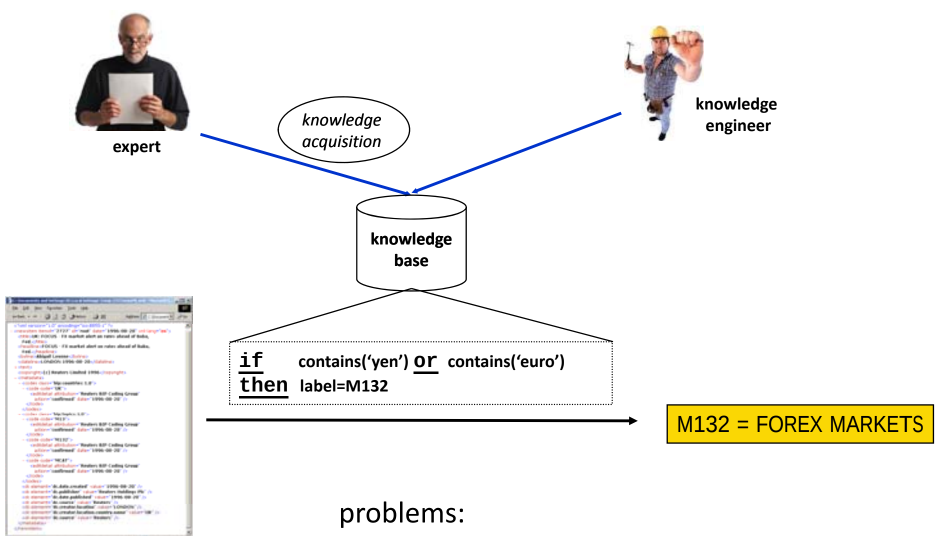
- topics, functions, genre, author, style, dichotc (e.g. spam/nom-spam), industry vertical, sentiment, language

taxonomies: international patent classification (IPC)

- IPC: section, class, subclass, group, subgroup ≈ **69,000**



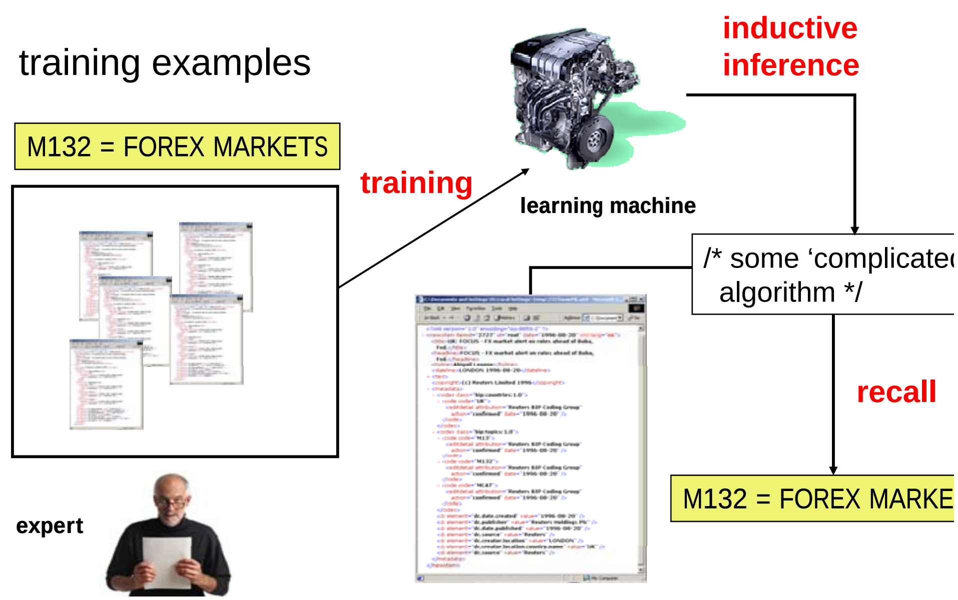
Solution (?): Explicit knowledge elicitation



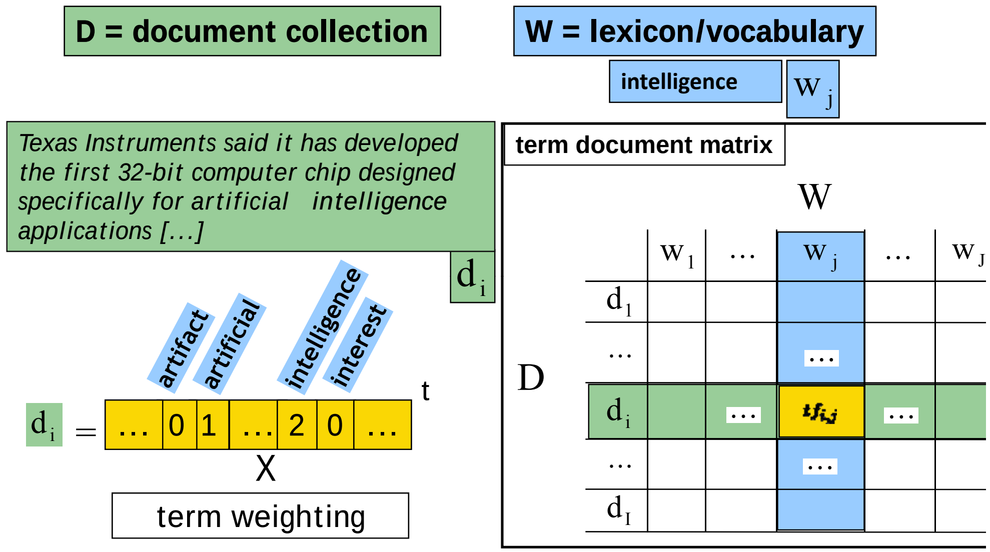
problems:

- low coverage
- moderate accuracy
- elicitation is often difficult and time-consuming

Solution (!): Example-based text categorization



Term document matrix & document vectors



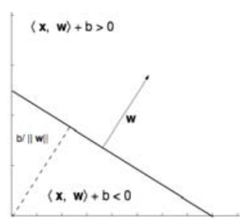
2. Supervised Classification

Binary Classification

- Each document is encoded as a feature vector
- Predict whether document belongs to a given category or not.
- Use linear classifier $f : \Re^d \rightarrow \{-1, 1\}, \quad f(\mathbf{x}) = \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$

Parameter

- Geometric view: separating hyper-planes



- Goal: minimize expected classification error

$$\mathbf{E}[y \neq f(\mathbf{x})] = \int \frac{(1 - y f(\mathbf{x}))}{2} dP(\mathbf{x}, y)$$

- Given: training set of labeled examples

$$\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$$

Perceptron Learning Algorithm

- Invented in the late 1950ies
- Extremely simple, yet powerful (extensions)
- Discarded by Minsky & Papert 1960ies
- Re-discovered in the 1990ies
- **Mistake driven** algorithm

```
w ← 0, b ← 0
repeat
  errors ← 0
  /* cycle through all training example
  for i = 1, ..., n do
    if sign(⟨w, x_i⟩ + b) ≠ y_i then
      w ← w + y_i x_i
      b ← b + y_i
      errors ← errors + 1
    end if
  end for
until errors = 0
```

Novikoff's Theorem

- Functional margin of a data point with respect to classifier

$$\gamma_i \equiv y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$$

(signed distance, if weight vector = unit length)

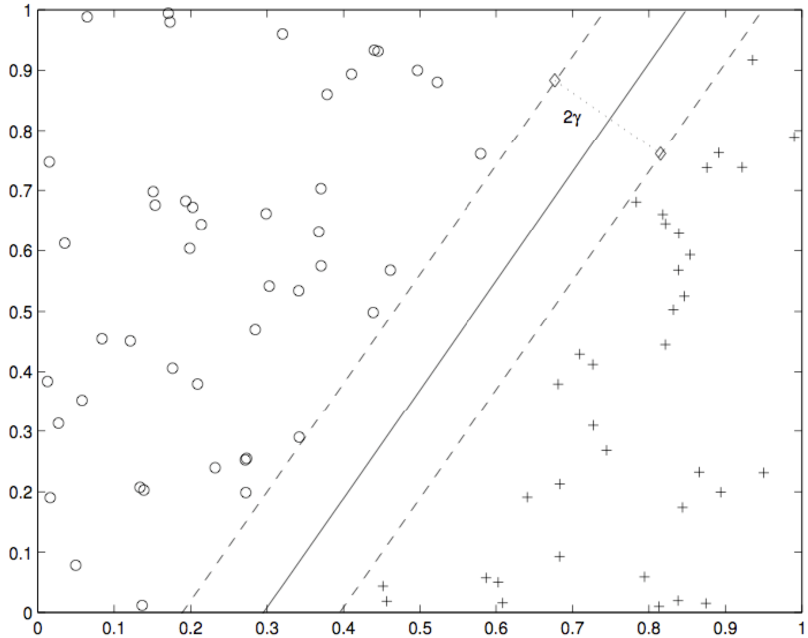
- Theorem:

Assume that there exists a weight vector $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ such that $\gamma_i = y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \geq \gamma$ for all examples.

Then the perceptron will not make more than $(R/\gamma)^2$ update steps.

(R is the radius of a data enclosing sphere)

Separation Margin



Novikoff's Theorem: Proof

- Lower Bound

$$\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle = \langle \mathbf{w}^{(t-1)}, \mathbf{w}^* \rangle + y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \geq \langle \mathbf{w}^{(t-1)}, \mathbf{w}^* \rangle + \gamma$$

$$\langle \mathbf{w}^{(t)}, \mathbf{w}^* \rangle \geq t\gamma$$

- Upper Bound

$$\begin{aligned} \|\mathbf{w}^{(t)}\|^2 &= \|\mathbf{w}^{(t-1)}\|^2 + y_i^2 \|\mathbf{x}_i\|^2 + 2y_i \langle \mathbf{w}^{(t-1)}, \mathbf{x}_i \rangle \\ &\leq \|\mathbf{w}^{(t-1)}\|^2 + \|\mathbf{x}_i\|^2 \leq \|\mathbf{w}^{(t-1)}\|^2 + R^2 \end{aligned}$$

$$\|\mathbf{w}^{(t)}\|^2 \leq tR^2$$

- Squeezing relations

$$\|\mathbf{w}^*\| \sqrt{t} R \geq \|\mathbf{w}^*\| \|\mathbf{w}^{(t)}\| \geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle \geq t\gamma \longrightarrow \quad t \leq R^2/\gamma^2$$

Compression Bound

Theorem:

For a fixed sample size m and $d \leq m$. Let $\mathbf{X}$ denote a sample set of size m drawn i.i.d. according to some unknown probability distribution $\mathbf{P}$. Assume based on $\mathbf{X}$ the perceptron algorithm converges after making mistakes on exactly d different examples. Then the probability that the generalization error of the obtained classifier is greater than ϵ is at most

$$\binom{m}{d}(1-\epsilon)^{m-d}$$

Proof of compression bound

- The probability of a classifier with generalization error $> \epsilon$ to classify n (i.i.d.) examples correctly is at most $(1-\epsilon)^n$.
- Fix a subset of examples $\mathbf{X}_d \subset \mathbf{X}$. Consider potential solutions $\mathbf{w}(\mathbf{X}_d)$ that classify $\mathbf{X}$ correctly, but that perform updates only on elements of $\mathbf{X}_d$. The probability for such a solution to be ϵ-bad is at most $(1-\epsilon)^{m-d}$.
- There are $\binom{m}{d}$ ways to select d examples from the m-sample $\mathbf{X}$, hence there are that many sets $\mathbf{X}_d$ and corresponding $\mathbf{w}(\mathbf{X_d})$.
- The perceptron solution is in one of the $\mathbf{w}(\mathbf{X}_d)$ sets. Don't know which one, but can apply the union bound.

Generalization Bound

- Generalization bound:

  Theorem: With probability more than $1-\delta$ over random draws of the sample $\mathbf{X}$ the following statement holds. If the perceptron algorithm converges by making mistakes on d examples, then the generalization error is less than

$$\frac{1}{m-d}\left[\log\binom{m}{d} + \log m + \log\frac{1}{\delta}\right]$$

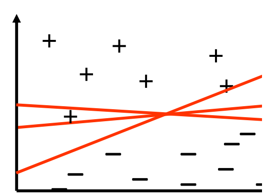
- The fewer mistakes are made in training, the better the guaranteed accuracy of the classifier.

Margin Maximization (Support Vector Machines)

- Separation margin (and sparseness) crucial for perceptron learning
- Idea: explicitly **maximize separation margin**

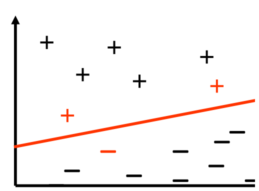
$$(\mathbf{w}^*, b^*) = \underset{(\mathbf{w},b),\|\mathbf{w}\|=1}{\operatorname{argmax}} \gamma$$

$$\text{such that } \forall i: \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq \gamma$$



- Reformulate as **quadratic program**

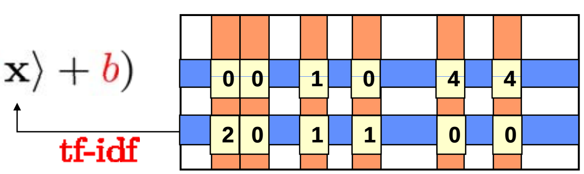
$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{s.t.} \quad y_i\left(\langle \mathbf{w}, \mathbf{x}_i\rangle + b\right) \geq 1, \quad \forall\, 1 \leq i \leq n$$



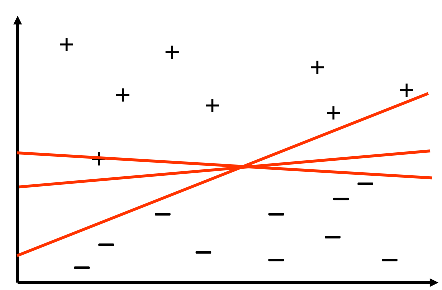
support vector machines

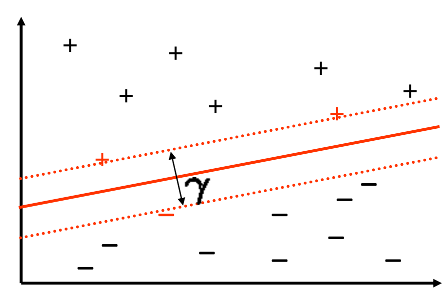
restriction to linear classifiers

$$f(\mathbf{x}) = \operatorname{sign}\left(\langle \theta, \mathbf{x}\rangle + b\right)$$



maximum margin principle





T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*, Kluwer, 2002

1. Text Categorization (cont'd)

Precision & Recall

	True label $+1$	True label -1
Predicted label $+1$	TRUE POSITIVE	FALSE POSITIVE
Predicted label -1	FALSE NEGATIVE	TRUE NEGATIVE
$\sum$	TOTAL POSITIVE	TOTAL NEGATIVE

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Experimental Evaluation

- Text categorization results:

microaveraged precision/recall breakeven-point [0..100]	Reuters	WebKB	Ohsumed
Naive Bayes	72.3	82.0	62.4
Rocchio Algorithm	79.9	74.1	61.5
C4.5 Decision Tree	79.4	79.1	56.7
k-Nearest Neighbors	82.6	80.5	63.4
SVM	**87.5**	**90.3**	**71.6**

- Machine Learning award 2009: most influential paper from 1999
  [Thorsten Joachims, ICML 1999]

- Much follow-up research …

Practical Use Cases

Google

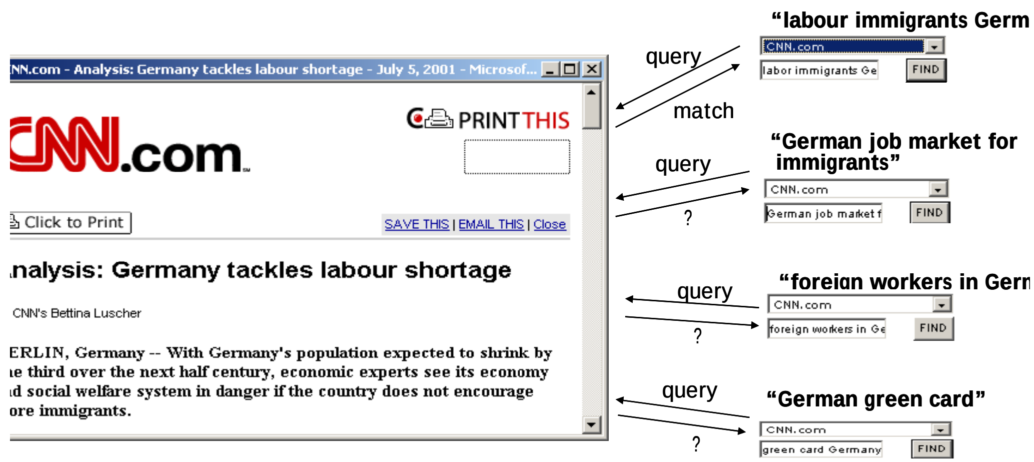
- label Web pages as child safe or not (for safe search)
- classifies billions of pages
- Many other features (other than text) used

Recommind

- Map documents to corporate taxonomy
- MindServer classification
- uses SVM light package

3. Semantic Search

Vocabulary mismatch problem



G. W. Furnas, T. K. Landauer, L. M. Gomez , S. T. Dumais, *The Vocabulary Problem in Human-System Communication: an Analysis and a Solution*. Bell Communications Research, 1987

Search as statistical inference

document in bag-of-words representation

China US trade relations

Search

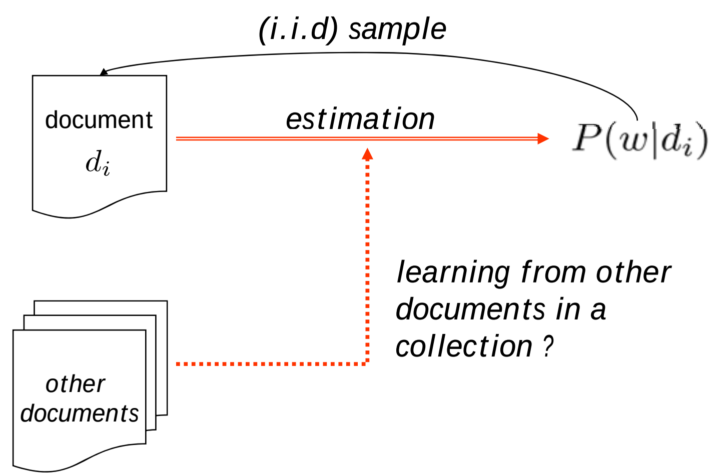
$$P(\text{'China'}|\text{all other words})$$

$$P(\text{'trade'}|\text{all other words})$$

How probable is it that terms like "China" or "trade" might occur?

automatically inferred key words c
be added to enrich document inde
document expansion

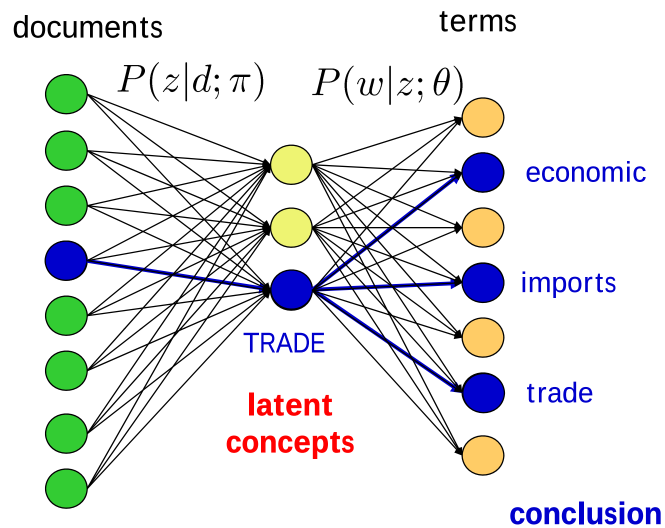
Estimation problem



- **crucial question**: In which way can the document collection be utilized to improve probability estimates?

4. Probabilistic Semantic Analysis

Estimation via probabilistic LSA



concept expression proba bilities are estimated bas on all documents that are dealing with a concept.

"unmixing" of superimpos concepts is achieved by statistical learning algorithm.

conclusion: $\Rightarrow$ no prior knowledge about concepts required, context an term co-occurrences are exploited

T. Hofmann, *Probabilistic Latent Semantic Analysis*, Uncertainty in Artificial Intelligence, UAI 1999.

pLSA - latent variable model

structural modeling assumption (**mixture** model)

$$\hat{P}(w|d) = \sum_{z=1}^{k} P(w|z;\theta)P(z|d;\pi)$$

document language model

latent concepts or topics

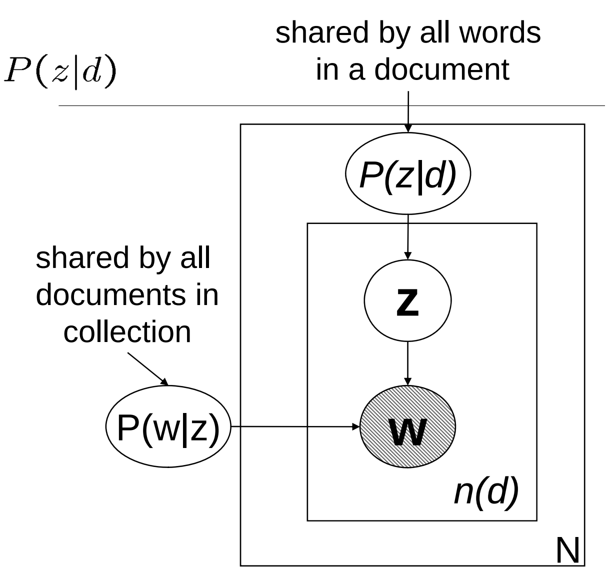
concept expression probabilities

document-specific mixture proportions

model fitting

pLSA - graphical model

$$P(w|d) = \sum_{z} P(w|z)P(z|d)$$

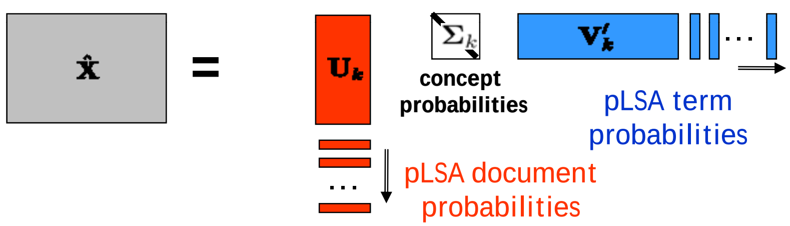


pLSA: matrix decomposition

mixture model can be written as a **matrix factorization**

equivalent symmetric (joint) model

$$\hat{P}(d,w) = \sum_{z=1}^{k} P(d|z)\, P(z)\, P(w|z) \qquad = P(d)\sum_z P(w|z)P($$



contrast to LSA/SVD: **non-negativity** and **normalization** (intimate relation to non-negative matrix factorization)

D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, NIPS 13, pp. 556-562, 2001.

pLSA via likelihood maximization

log-likelihood

$$L(\theta,\pi;n) = \sum_{d,w} \underbrace{n(d,w)}_{\text{observed word frequencies}} \log \underbrace{\left[\sum_z P(w|z;\theta)P(z|d;\pi)\right.}_{\hat{P}(w|d)}$$

$$\xrightarrow{\text{argmax}} (\hat{\theta},\, \hat{\pi})$$

predictive probability of pLSA mixture model

goal: find model parameters that maximize the log-likelihood, i.e. maximize the average predictive probability for observed word occurrences (**non-convex problem**)

"The meaning of a word is its use in the language".
- Ludwig Wittgenstein, Philosophische Untersuchungen

Expectation maximization algorithm

E step: posterior probability of latent variables ("concepts")

$$P(z|d,w) = \frac{P(z|d;\pi)P(w|z;\theta)}{\sum_{z'} P(z'|d;\pi)P(w|z';\theta)}$$

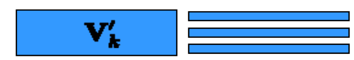
Probability that the occurrence of term w in document d can be "explained" by concept z

M step: parameter estimation based on "completed" statistics

$$P(w|z;\theta) \propto \underbrace{\sum_d n(d,w)P(z|d,w)}_{\text{how often is term } w \text{ associated with concept } z\,?}, \qquad P(z|d;\pi) \propto \underbrace{\sum_w n(d,w)P(z|d}_{\text{how often is document associated with concept}}$$

A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of Statistical Society B, vol. 39, no. 1, pp. 1-38, 1977

Example



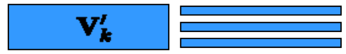
concepts (3 of 100) extracted from AP news

Concept 1	
securities	94.96324
firm	88.74591
drexel	78.33697
investment	75.51504
bonds	64.23486
sec	61.89292
bond	61.39895
junk	61.14784
milken	58.72266
firms	51.26381
investors	48.80564
lynch	44.91865
insider	44.88536
shearson	43.82692
boesky	43.74837
lambert	40.77679
merrill	40.14225
brokerage	39.66526
corporate	37.94985
burnham	36.86570

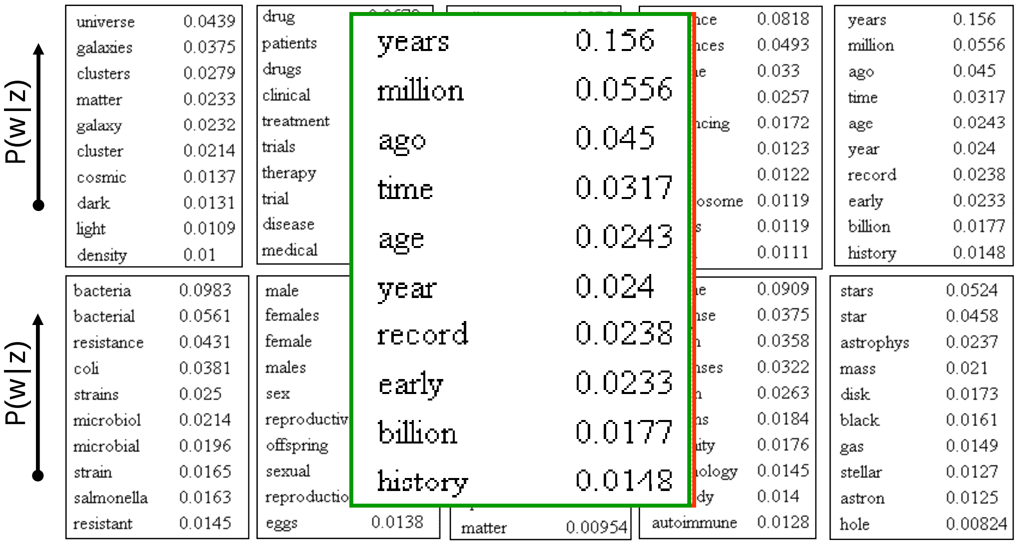
Concept 2	
ship	109.41212
coast	93.70902
guard	82.11109
sea	77.45868
boat	75.97172
fishing	65.41328
vessel	64.25243
tanker	62.55056
spill	60.21822
exxon	58.35260
boats	54.92072
waters	53.55938
valdez	51.53405
alaska	48.63269
ships	46.95736
port	46.56804
hazelwood	44.81608
vessels	43.80310
ferry	42.79100
fishermen	41.65175

Concept 3	
india	91.74842
singh	50.34063
militants	49.21986
gandhi	48.86809
sikh	47.12099
indian	44.29306
peru	43.00298
hindu	42.79652
lima	41.87559
kashmir	40.01138
tamilnadu	39.54702
killed	39.47202
india's	39.25983
punjab	39.22486
delhi	38.70990
temple	38.38197
shining	37.62768
menem	35.42235
hindus	34.88001
violence	33.87917

Example

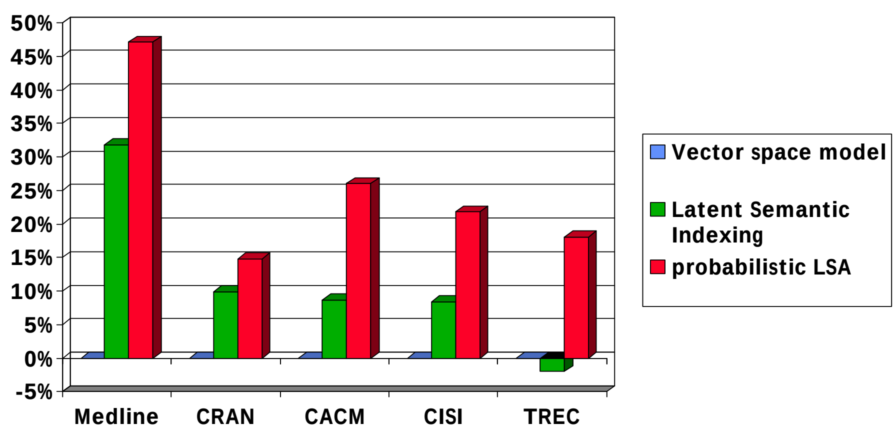


concepts (10 of 128) extracted from science magazine articles (12K)



3. Semantic Search (cont'd)

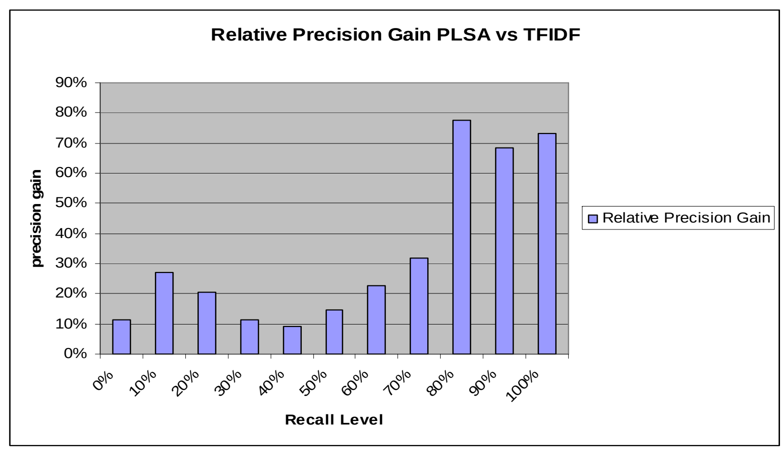
Experimental evaluation



15-45% relative improvement gain in precision compared to SMART retrieval metric

Experiments - TREC3 (AP collection)

comparison with TF-IDF metric (SMART) on TREC3



pLSA algorithms achieved a mean average precision (MAP) gain of 20%, in particular in the high recall range

Practical Use Cases

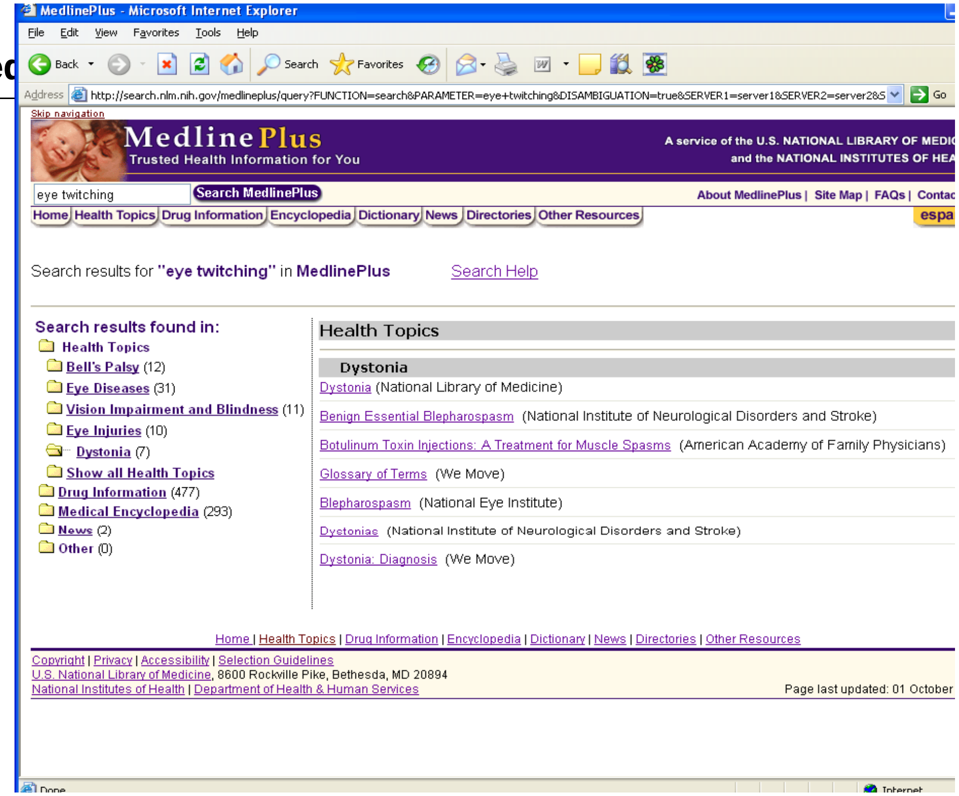
Google

- Somewhat similar model used to extract concepts from documents
- Used to improve search result ranking

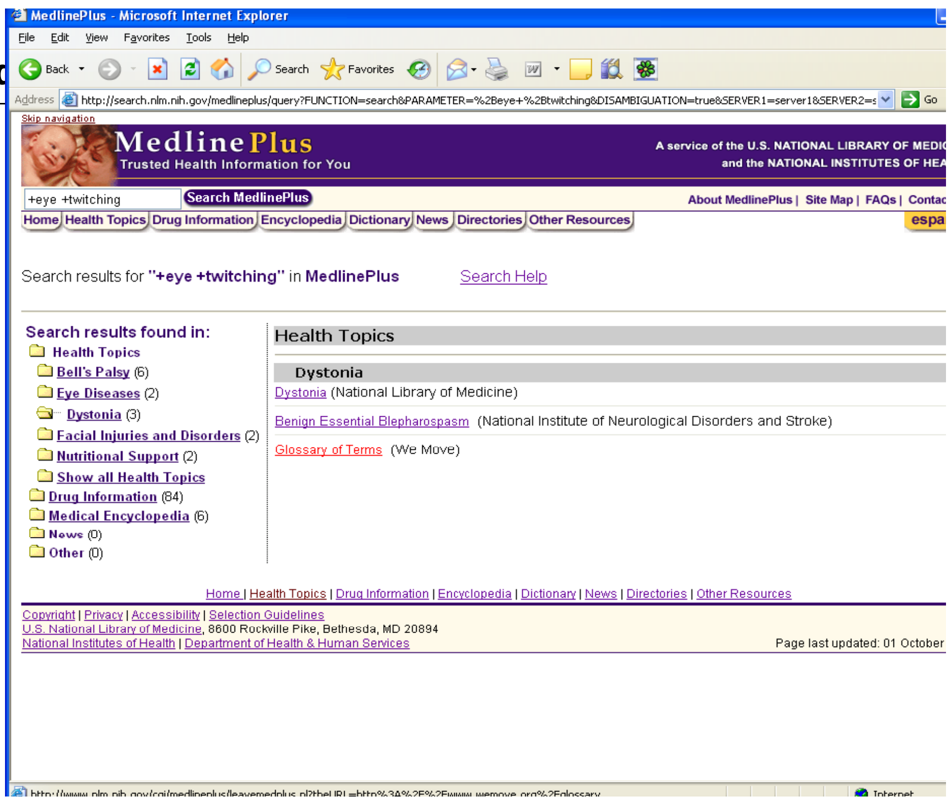
Recommind

- Heart of intelligent retrieval systems
- Many customers: Medline, law firms, enterprise search
- Allows to learn aspects of relevant semantics of domain purely based on co-occurrence statistics

Med



Med



5. Ranking

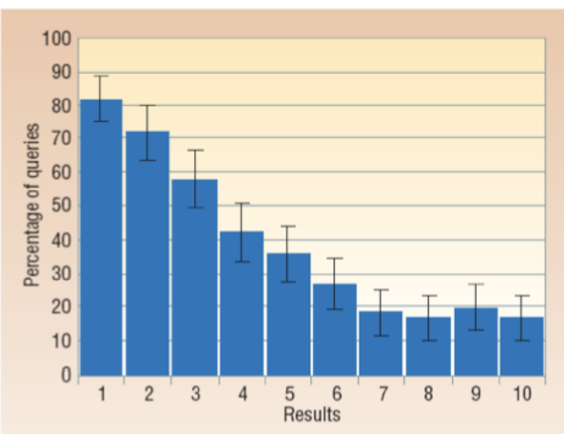
Learning for Ranking: History

- Bollmann & Wong 1987 [BW87], Wong & Yao 1988 [WY88]
  - Acceptable ranking as one that respects known pairwise preferences
  - Perceptron algorithm to learn ranking function
- Fuhr 1989 [Fuh89]
  - Polynomial basis functions to map document-query representations to (known) relevance probabilities
  - Least squares regression
- Bartell et al. 1994 [BCB94]
  - Combination of different rankings (aka experts, meta search)
  - Linear expert score combination
  - Gutman's point alineation as a measure of rank correlation
  - Conjugate gradient descent optimization

Relative Relevance from Result Clicks

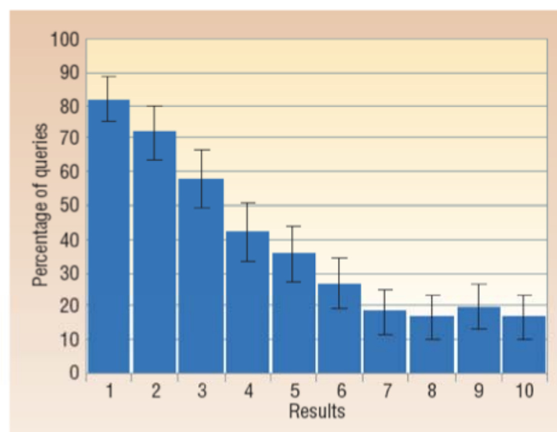
- Clicks do not imply absolute relevance judgments
- Scanning order: a click at rank k implies result is better than non-clicked on $k' < k$
- Agnostic with regard to results below last click
- T. Joachims: Optimizing Search Engines Using Clickthrough Data [Joa02]

What Users Look at: Eye Tracking Experiments



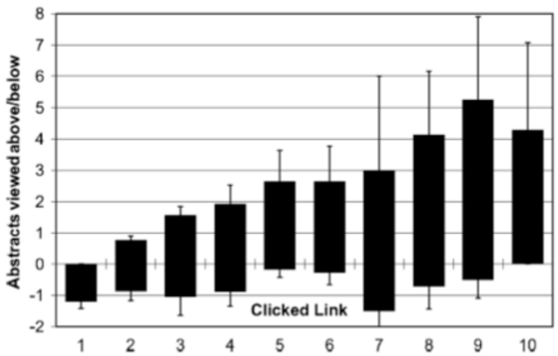
- % of queries where a user viewed result presented at particular rank
- Result beyond rank 3 examined $< 50\%$ of times
- Short attention span and skewed towards top positions
- Source: Joachims & Radlinski [JR07]

Habitual Judgment Bias



- % of queries where a user viewed result presented at particular rank
- Result beyond rank 3 examined $< 50\%$ of times
- Short attention span and skewed towards top positions
- Source: Joachims & Radlinski [JR07]

Scanning



- Mean number of snippets viewed before click on rank r
- Only about 1 snippets examined following the clicked rank.
- # examined snippet increases with rank – not quite linearly (skips).
- Source: Granka, Joachims & Gay [GJG04]

Attention Modeling: Findings

- Users only examine a small portion of the information displayed on a result page
- In result list view, the top results get dis-proportionately more attention
- Users typically sequentially scan the result list from top to bottom (with possible skips).
- Acquired habits and experience with search engines influence attention and click probability

6. Learning to Rank

Relative Relevance Feedback from Result Clicks

- Extraction of pairwise preferences, $u_i \prec u_j$ is a URL u_i is preferred over u_j with regard to a (fixed) query q
- For a query q and a result ranking of URLs $(u_1, u_2, u_3, \dots)$ with click variables $c_i \in \{0, 1\}$, $u_i \prec_q u_j$ if and only if $i > j$ and $c_i = 1 \wedge c_j = 0$.
- Conservative preference extraction
- No absolute relevance assessment per query-URL pair

$\Rightarrow$ learning algorithms using pairwise preference training data

Kendall's tau

- Kendall's τ: simlarity measure for rankings $\pi, \pi' \in \mathcal{S}_n$
- P = concordant pairs (i.e. URLs u_i and u_j with $i = j$ ordered in the same manner by π and π'.
- Q = disconcordant pairs
- Definition

$$\tau(\pi, \pi') = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{n}{2}}$$

- Example

$$\pi = (1, 2, 3, 4, 5)$$
$$\pi' = (3, 2, 1, 4, 5)$$
$$\tau(\pi, \pi') = \frac{7 - 3}{7 + 3} = 0.4$$

Kendall's tau

- Fulfills axioms of Kemeny & Snell
- In case of binary relevance, related to average precision:

$$AvPrec(\pi) = \frac{1}{R}\left[Q + \binom{R+1}{2}\right]^{-1}\left(\sum_{i=1}^{R}\sqrt{i}\right)^2$$

where R is the number of relevant documents (cf. [Joa02])
- Learning goal: Minimize expected τ over query/ranking pairs

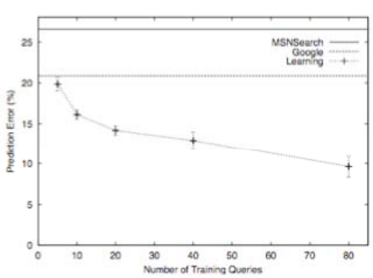
$$\mathbf{E}_P[\tau(f)] = \int \tau(\pi^*, \pi_{f(q)}) dP(q, \pi^*)$$

SVM Ranking for Pairwise Preferences

- Basic idea: enforce a separation margin on each know preference pair
- Formally (with slack variables):

$$\langle \mathbf{w}, \Phi(q, u_i) - \Phi(q, u_j) \rangle \geq 1 - \xi_{i,j,q}, \quad \forall i, j: \; u_i \prec_q u_j$$

- For given q, URLs u can be ranked according to $\langle \mathbf{w}, \Phi(q, u) \rangle$.



weight	feature
0.60	query_abstract_cosine
0.48	top10_google
0.24	query_url_cosine
0.24	top1count_1
0.24	top10_msnsearch
0.22	host_citeseer
0.21	domain_nec
0.19	top10count_3
0.17	top1_google
0.17	country_de
...	
0.16	abstract_contains_home
0.16	top1_hotbot
...	
0.14	domain_name_in_query
...	
-0.13	domain_tu-bs
-0.15	country_fi
-0.16	top50count_4
-0.17	url_length
-0.32	top10count_0
-0.38	top1count_0

- Metasearch engine
- Example features
- Source: [Joa02]

Features used for ranking

What features should be used in the Φ function?

- should describe the match between a document d and a query q
- examples
  - number of words shared by query and document
  - number of shared words inside certain HTML tags
  - cosine similarity between query and document title or abstract
  - page rank of document d
  - rank of d in the result list of q for some search engine (e.g. within top10, within top50, etc.)
  - properties of the URL (contains tilde, length, etc.)
  - ...

Learned Ranking

Comparison	more clicks on learned	less clicks on learned	tie (with clicks)	no clicks	total
Learned vs. Google	29	13	27	19	88
Learned vs. MSNSearch	18	4	7	11	40
Learned vs. Toprank	21	9	11	11	52

weight	feature
0.60	query_abstract_cosine
0.48	top10_google
0.24	query_url_cosine
0.24	top1count_1
0.24	top10_msnsearch
0.22	host_citeseer
0.21	domain_nec
0.19	top10count_3
0.17	top1_google
0.17	country_de
...	
0.16	abstract_contains_home
0.16	top1_hotbot
...	
0.14	domain_name_in_query
...	
-0.13	domain_tu-bs
-0.15	country_fi
-0.16	top50count_4
-0.17	url_length
-0.32	top10count_0
-0.38	top1count_0

- Features have been (roughly) hand designed
- Numbers indicate weights learned by ranking SVM

Applications

- How can the learning from clickthrough data approach be applied?
  - clickthrough can not be used immediately to improve search results for a specific query
  - preferences can be aggregated over the **whole user population** to **self-optimize a parameterized ranking function**
  - optimization can also be performed for **specific groups of users** (e.g. users from the same country) to construct **adaptive and personalized ranking functions**
- In particular:
  - **meta-search engine**: combine results from different search engines (e.g. parameterized ranking function corresponds to combination of search results)

7. Summary

Machine Learning for Search

Text categorization:

Experts label documents, computers learn to generalize to new documents -> scalability & automation

Semantic search:

Statistical models learned from document corpus help bridge the semantic gap in search -> more relevant search results

Learning to rank:

Users provide implicit feedback through clicks that help improve the result ranking.

Many more, related applications … Many more methods …

The future: **intelligent Web**

- use of social intelligence and machine learning.