

Intelligent Internet Information Retrieval

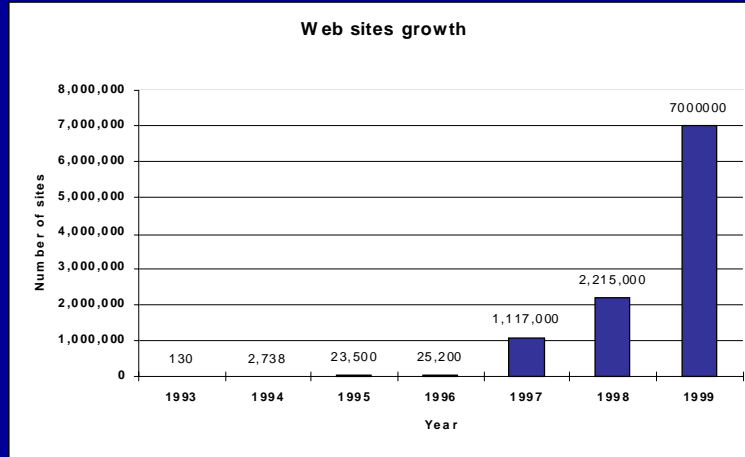
Yu Hen Hu
University of Wisconsin-Madison
Dept. Electrical and Computer Engineering
hu@engr.wisc.edu

IBIQR work is collaborated with Prof. Jong-Min Park @San Diego State U.

Outline

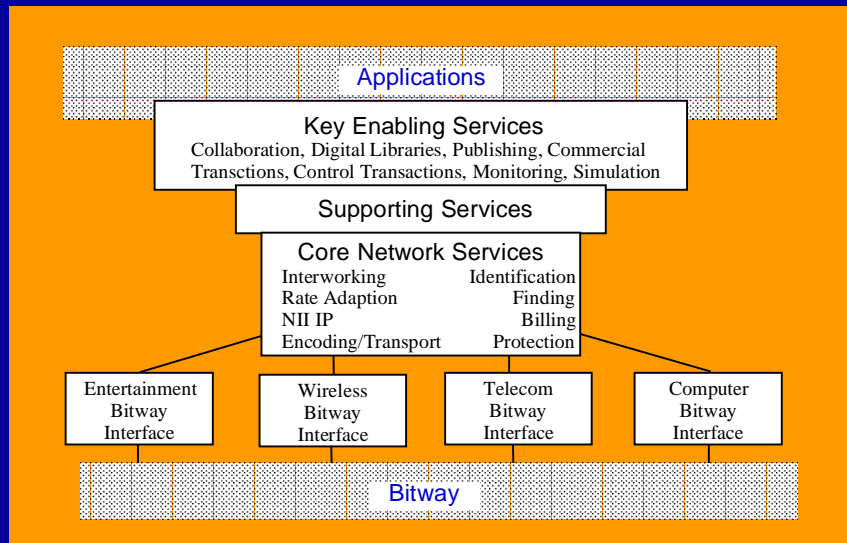
- State of Internet
- Internet Information Retrieval Process
- Problems with current Internet Information Retrieval
- An example: IQBIR (Intelligent Query and Browsing Information Retrieval)

State of Internet



source: Hobbes Internet Timeline

NII Services Model



Key Enabling Services

- *Collaboration*: 1 to 1, 1 to many, many to many interaction. Examples: teleconferencing, virtual organization (classroom, corporation), interactive TV
- *Digital Library*: public read-only access to electronic publications
- *Publishing*: to create and distribute publications electronically. E.g. electronic newspaper, pay-per-view, movie on demand
- *Commercial Transactions*: EDI, digital cash, virtual mall
- *Control Transactions*: distributed manufacturing, production, process control, travel and other reservation, etc.
- *Monitoring*: remote monitoring, patient care, etc.
- *Simulation*: virtual reality, network games, remote presence, etc.

Supporting Services

Services which support key enabling services. Examples are:

- **Specialized information management services** for the storage, distribution, retrieval, and translation of information. e.g.
 - Knowledge based query processing
 - Knowledge discovery: content based indexing, processing
 - Knowledge based language translation
- Industry specific services: e.g. Electronic Data Interchange (EDI)
- Public (government sector) services
- Competitive version of core networking services: e.g. On-line
Yellow page on non-local services
- Remote Pay-Per-Use Services
- Service support mechanisms

What Information on the Web?

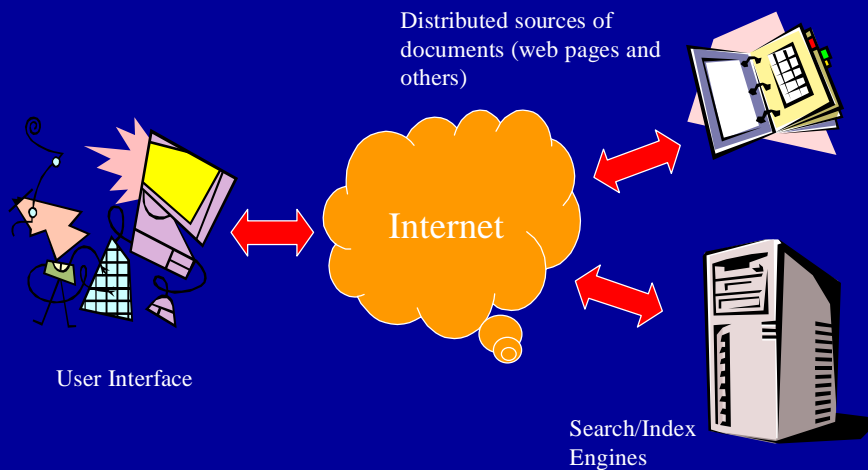
- 800M web pages, 6 Tbytes text data and 180M images for Tbyte image data on 3M publicly indexable web servers -- 2/99 sample results.
- 6 public search engines (Alta Vista, Excite, HotBot, Infoseek, Lycos, and Northern Light) collectively covers 60% of the web, with maximum about 16%.
- 83% contents are commercial, followed by 6% science and education, 2.8% health, 2.5% personal, 1-2% societies, pornography, community, government, and religion.

Nature, 7/99, pp107-9www.nature.co

Internet Search



Internet Information Retrieval

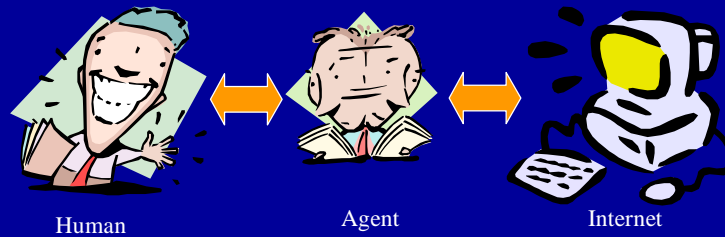


Issues in Internet Search

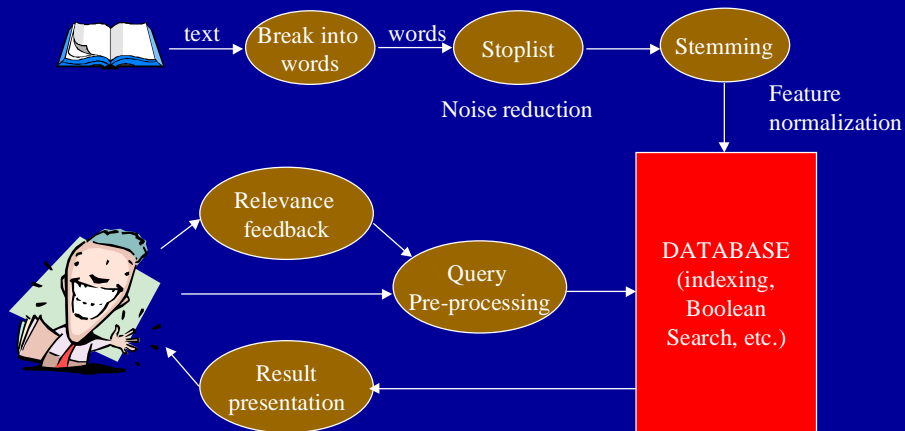
- Precision
WYGIWYW: What you get is what you want?
- Recall
Are all the available information sources searched?
- Efficiency
How soon the desired search results can be obtained.

Active Learning Agent for Internet Search

- A mediator (agent) between human user and search engine, performing tasks to
 - Interpret human queries,
 - Organize search results,
 - Solicit relevance feedback from users (Asking questions)



Document Retrieval Process



IR = PC

- Information retrieval = Pattern classification
- Given a set of feature vectors $\{x_i\}$, each represents a document.
- A user provides a query, x^* , also in the form of a feature vector.
- Use x^* as a “prototype”, for a given error bound ϵ , label all documents as *relevant* if the corresponding feature vectors satisfy

$$d(x_i, x^*) \leq \epsilon$$

Otherwise, the document is labeled as *irrelevant*.

Query Structure

- Boolean Query:
 - A SET of terms joined by Boolean logic operators (AND, OR, NOT)
 - Natural to specify by user. E.g. “Beijing AND duck”
 - Easy to represent complicated queries but difficult to assign weight to different terms, and prone to user error
- Vector Query:
 - A term vector consisting of 0s and 1s at corresponding term location.
 - Difficult to specify explicitly by user. Needs a translator module.
 - Easy to weight different terms, but difficult to describe complicated relations

Boolean Query

- List of terms (keywords) joined using Boolean operators.
- Can be represented in conjunctive normal form (CNF) or disjunctive normal form (DNF)
 - Conjuncts: terms or terms joined by AND
 - Disjuncts: conjuncts joined by OR
- Example:
 - soccer and not football
 - Beijing or Shanghai
 - (round and orange) or (square and yellow): DNF
 - (round or square) and (orange or yellow): CNF
 - 2 of (A, B, C)
- Contingency table for query match

Boolean Matching: Contingency Table

		Document	
		Term in	Term not in
Q u e r y	Term in	Match	Not match
	Term not in	Not match	Not match

Term Vector Query

- Each document is represented by a specific *term vector*
- A *term* is a key-word or a phrase
- A *term vector* is a *vector of terms*. Each dimension of the vector corresponding to a term.
- Dimension of a term vector = total number of distinct terms.
- Example:
 - Set of terms = [tree, cake, happy, cry, mother, father, big, small]
 - document = “Father gives me a big cake. I am so happy”, “mother planted a small tree”
 - Term vectors: [0, 1, 1, 0, 0, 1, 1, 0], [1, 0, 0, 0, 1, 0, 0, 1]

Term Weighting

- All terms are NOT created equal!
- f_{ik} = # of times term k appeared in document d_i .
- $t_k = \sum f_{ik}$ = total number of term appearance in all documents.
- $p_{i,k} = f_{ik}/t_k$ = Prob. term k appears in document d_i .
- $h_k = -\sum_i p_{i,k} \log_2 p_{i,k}$ = entropy of term k .
 $h_k = 0$ if only one $p_{i,k} = 1$ and the rest = 0. Otherwise, $h_k > 0$.
- Term weight
 - $s_k = \log_2 t_k - h_k$
 - s_k is large when t_k is large (term appears frequently), and when h_k is small (term appears in few documents)
- Drawbacks:
 - If term k appears only once in one document, and none others, ($p_{i,k} = 1, t_k = 1$), $s_k = 0!$

Term Weight Example

f(i,k)	k=1	2	3	4
doc. A	2	1	0	3
doc. B	1	1	1	1
doc. C	0	1	0	0
tk	3	3	1	4

p(i,k)	k=1	2	3	4
a	0.667	0.333	0.000	0.750
b	0.333	0.333	1.000	0.250
c	0.000	0.333	0.000	0.000
hk	0.918	1.585	0.000	0.811
sk	0.667	0.000	0.000	1.189

Note that $s_2 = s_3 = 0$ although their corresponding terms are distributed quite differently.

Inverse Term Frequency Vector

- A probabilistic term vector representation.
- Relative Term Frequency (within a document)
 - $t_f(t,d) = \text{count of term } t / \# \text{ of terms in document } d$
- Inverse document Frequency
 - $d_f(t) = \text{total count of document} / \# \text{ of doc contain } t$
- Weighted term frequency

$$d_t = t_f(t,d) \cdot \log [d_f(t)]$$

- Inverse document frequency term vector $D = [d_1, d_2, \dots]$

ITF Vector Example

Document 1: The weather is great these days.

Document 2: These are great ideas

Document 3: You look great

Eliminate: The, is, these, are, you

Term	tf(t,1)	tf(t,2)	tf(t,3)	df(t)	D1	D2	D3
Weather	1/6	0	0	3	0.08	0.00	0.00
grea	1/6	1/4	1/3	1	0.00	0.00	0.00
day	1/6	0	0	3	0.08	0.00	0.00
idea	0	1/4	0	3	0.00	0.12	0.00
look	0	0	1/3	3	0.00	0.00	0.16

Keywords (Feature) Normalization

- Stop-list:
 - Words and phrases that can not help discriminating one document from others should be excluded. A dictionary of these words, phrases, called *stop-list* may be used.
- Stemming:
 - Each word may have plural or other forms. Selecting unique form of each word is called *stemming*
- Thesauri:
 - Keywords of the same meaning may be grouped into one representative term.

Stemming

- **Affix removal:** removing suffixes and prefixes from terms, leaving a *stem*.
- **Successor variety:** uses frequencies of letter sequences in a body of text as basis of stemming.
- **Table look-up**
- **n-gram:** conflates terms based on n-grams they share. Word clustering method.
- **Examples**
 - skies -> sky not ski!
 - fed -> feed
 - hopping -> hop, but falling -> fall
 - sensitivity -> sensitive, but sensibility -> sensible

Porter, M.F. 1980 "An algorithm for suffix stripping", *Program*, 14(3),130-7.

Thesauri

- *A thesaurus*
 - provides a precise and controlled vocabulary.
 - Specifies relations among terms:
 - equivalence: A.K.A., used for, use
 - hierarchy: part of, contains
 - other relations: e.g. property
 - Domain (application) dependent
- **Constructing a thesaurus**
 - Manual construction
 - Automated construction
 - From a collection of documents
 - Merging existing thesauri
 - User specific thesauri
- **References**
 - Information Retrieval, W. B. Frakes & R. Baeza-Yates (Eds), Prentice Hall, 1992 (ISBN 0-13-463837-9)
 - Information storage and retrieval, R. R. Korfhage, John Wiley & Son, 1997, (ISBN 0-471-14338-3)

Document Matching Criteria

- A *pattern classification* problem formulation:

Given a query vector q and a collection of documents, represented by corresponding term vectors $\{d_i\}$, devise a document matching criterion to label each d_i as either $r=1$ (relevant) or $r=0$ (irrelevant) such that the probability of misclassification is minimized.

- A simple matching rule:

Define a *distance measure* between q and d_i , $d(q, d_i)$. Find a threshold h such that

d_i is relevant ($r=1$) if $d(q, d_i) \leq h$
 d_i is irrelevant ($r=0$) if $d(q, d_i) > h$

- The distance measure can be defined on different norms.

Distance Measure

- L-Norm distance

$$d(q, d_i) = \|W(q - d_i)\|_L$$

W : diagonal weight matrix.

$L = 2$: Euclidean distance

- Cosine distance

$$d(q, d_i) = q^T d_i / \{|q| \cdot |d_i|\}$$

cosine of angle between q and d_i

- Hamming distance

$$d(q, d_i) = \sum_{m=1}^M w_m (q_m \oplus d_{i,m})$$

w_m : m -th diagonal element of W

q_m : m -th element of q

$d_{i,m}$: m -th element of d_i

\oplus : exclusive OR operator

both are either Boolean variables 0 or 1.

- These measures can also be used for clustering documents.

Statistical Pattern Classification Formulation

- Given a particular query q , a randomly drawn document d_i is relevant to q with the *prior probability* $p_r = p\{r|q\}$. Define
- Likelihood: $p\{d_i|r, q\}$
- Marginal prob.: $p\{d_i|q\}$
- Posterior prob.: $p\{r | d_i, q\}$
- Bayes rule:

$$p\{r | d_i, q\} = \frac{p_r \cdot p\{d_i | r, q\}}{p\{d_i | q\}}$$
- Maximum a posterior (MAP) classifier:

Assign $r = 1$ to d_i if

$$p\{r=1|d_i, q\} > p\{r=0|d_i, q\}$$

Otherwise, assign $r = 0$.
- Discussion:
 - $p\{r|d_i, q\}$ is very difficult to estimate directly.
 - *Prior probability* and *likelihood* function may be easier to estimate and to model.
 - All inferences depend on q .

Pattern Classifiers

- Distance Based Classifiers
 - Nearest neighbor (NN)
 - K-nearest neighbors (KNN)
 - Clustering based nearest neighbor methods such as Learning vector quantization (LVQ)
- Model Based Classifiers
 - Maximum likelihood
 - Maximum A Posterior
- Discriminant Classifier
 - Linear hyperplane classifier
 - Multi-layer perceptron classifier
 - Decision tree classifier
 - Support vector machine
- For IR application, distance based classifier is easier to apply.

Dependence on Query

- Query is often specified with only terms that need to be included in documents. The rest are considered *don't cares*.
- To implement don't cares, one may project each d_i into a subspace, denoted by e_i such that $e_{i,m} = 0$ if $q_m = 0$.
- For query-specific distance measures, replace d_i by e_i and then calculate the corresponding distances.
- Perform statistical pattern classification, perform the same mapping, and then remove all dependency on q from the resulting expression to yield: $p\{r\}$, $p\{r|e_i\}$, $p\{e_i|r\}$, and $p\{e_i\}$.
- However, these probabilities are still specific to each different query. In practice, it is not convenient to derive them for each query.

User Profiles

- Purpose
 - Provide user preferences regarding specific types of query or all queries in general.
- Target
 - Single user
 - Group of users.
- Format
 - Specific term weights
 - Specific term vectors
- Direct Method
 - Gather information (term weights, frequently used terms, as well as personal information such as age, sex, profession) directly from user.
- In-direct Method
 - Monitor and archive user queries. Then use data mining methods such as clustering to discover regular patterns.
- Privacy issue must be taken into account

Query Modifications

- Users often do NOT know exactly the term used in the IR system to retrieve what they want. Thus,
- Initial query needs to be modified to improve the chance of high precision Information retrieval.
- However, any modification must seek user approval.
- Normalization:
 - removing stop words
 - stemming
 - replacement with equivalent standardized index terms using thesauri
- Re-weighting
 - Using user profiles
- Expansion:
 - adding relevant terms using thesauri

Other Forms of Queries

- Natural language query
 - Users provide complete sentence as a form of query
 - Natural language understanding modules are used to parse the sentence to extract
 - key words and ...
 - contextual information (difficult if possible)
 - Practical only in an controlled environment such as dialog query
- Dialog query
 - IR system is allowed to work with user to form the query interactively.
 - Conversation can be in natural language format
 - Asking context-sensitive questions depending on previous conversations with the user.

Performance Evaluation

- Document retrieval problem is a hypothesis testing problem:

H_0 : d_i is relevant to q ($r=1$)

H_1 : d_i is irrelevant to q ($r=0$)

- Type I error ($P_{e1} = P\{r=0|H_0\}$)
Relevant but not retrieved.
- Type II error ($P_{e2} = P\{r=1|H_1\}$):
Irrelevant but retrieved.

Contingency table for evaluating retrieval

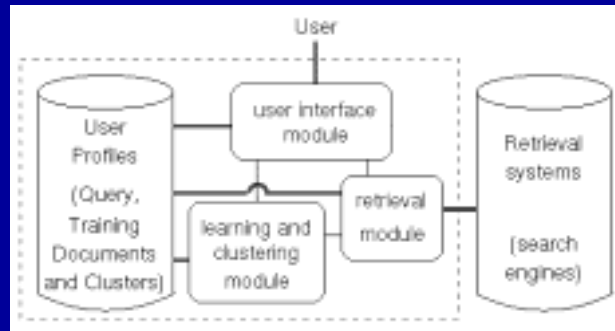
	Retrieved	Not retrieved
Relevant	w	x
Irrelevant	y	z

- Precision Recall Curve
 - $P(\text{recision}) = w/(w+y)$ is a measure of *specificity* of the result
 - $R(\text{ecall}) = w/(w+x)$ is an indicator of *completeness* of the result.
- Operating curve
 - $P_{e1} = x/(w+x) = 1 - R$
 - $P_{e2} = y/(y+z) = F(\text{allout})$
- Expected search length = average # of documents need to be examined to retrieve a given number of relevant documents.
- Subjective criteria

Relevance Feedback

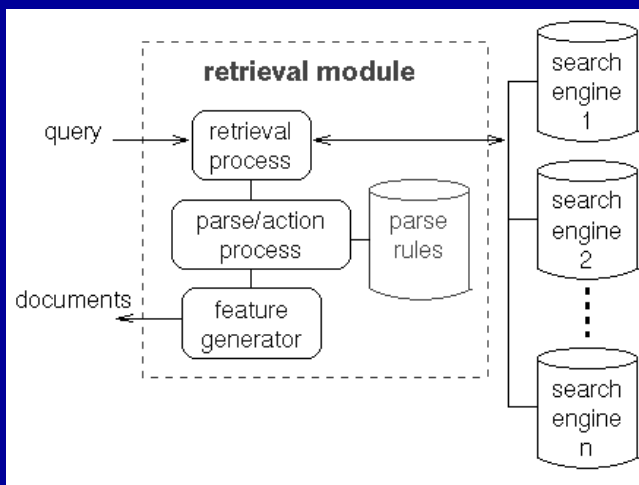
- User provide feedback on whether one or one group of retrieved documents are relevant to the query.
- A post-search dialogue. An extension of pre-search dialogue.
- Feedback can be used to
 - refine query
 - build user profile
- User Interface Problem: Users do NOT want to answer too many questions. (period)
- Issue: How to minimize amount of user feedback while achieving desired performance of precision and recall.

IQBIR System Architecture



Intelligent Query, Browsing Information Retrieval Agent

Meta-Internet Search Agent



Vocabulary and Parsing Rules

term index	character string	term index	weight
1	term 1	3	0.15
2	term 2	7	0.02
	⋮	23	0.12
	⋮	31	0.11
n	term n	32	0.30

(a) Vocabulary

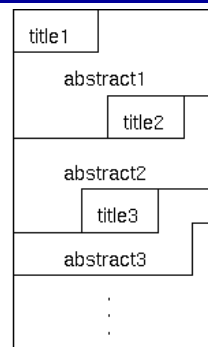
(b) example of a document vector

rule #	context	token	action	next rule
1	free	"of"		3
2	*	*		1
3	tagstart	"p"	DocStart	8
4	free	number	Size	11
⋮	⋮	⋮	⋮	⋮
8	tag	"href"	URLs	14
⋮	⋮	⋮	⋮	⋮

Web Document Database

URL1	title1 offset/length	abstract1 offset/length
URL2	title2 offset/length	abstract2 offset/length
URL3	title3 offset/length	abstract3 offset/length
⋮	⋮	⋮

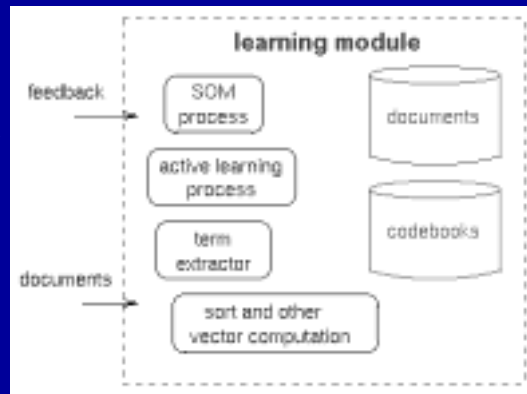
(a) Web document database



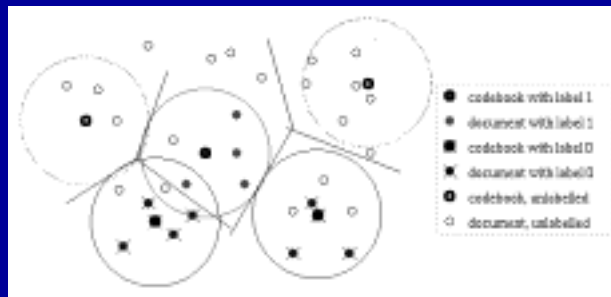
(b) Document corpora

Learning Module

- Use Self-organizing map to cluster documents into 16 clusters and 1 misc.. cluster.
- Use LVQ and active learning to classify documents into relevant, irrelevant, and don't care.

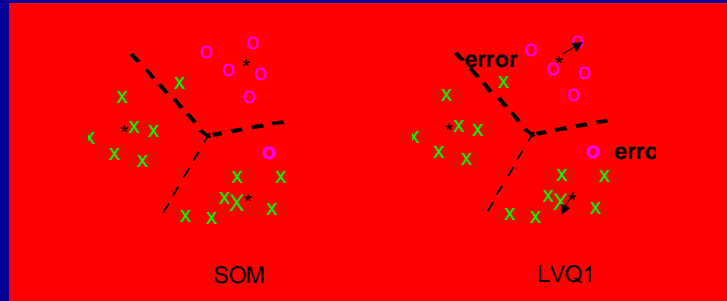


Active Sampling & Clustering



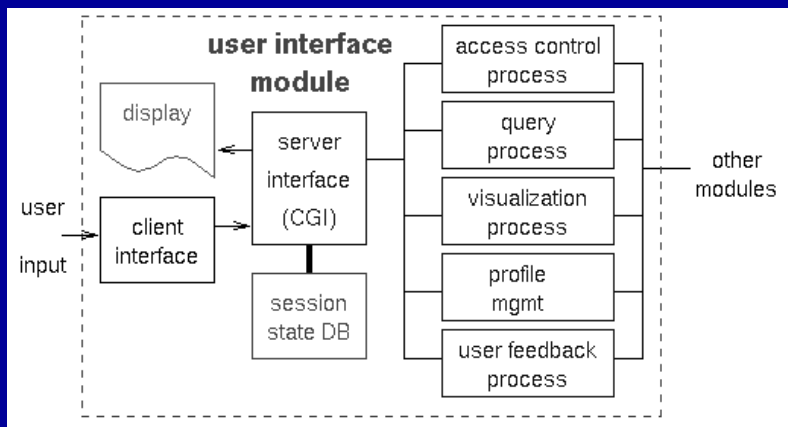
After clustering, fine-tune class boundary by sampling along decision boundary.

LVQ

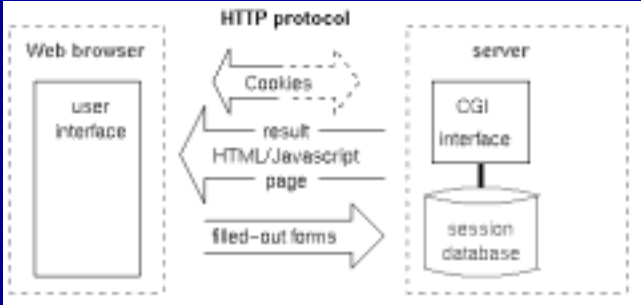


- LVQ adjust clustering centers (*) to correct mis-classification after initial clustering
- Corrections are made on mis-classified samples NEAR Decision Boundary

User Interface Module



Interface Module and User Profile



Internal Databases

User Profile
 Initial supplied query
 Last reformulated query
 Cluster codebooks
 Documents, including representatives
 Miscellaneous user tweakable parameters

Search Engines
 Query URL string
 First page access string
 Subsequent page access string
 Page number offset
 Parsing rules

Performance Evaluation Criteria

Precision and Recall ratios [Rijsbergen 79]

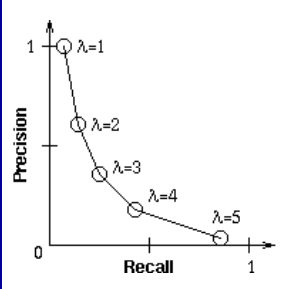
A: set of retrieved documents
 R: set of relevant documents

$$P = |A \cap R| / |R|$$

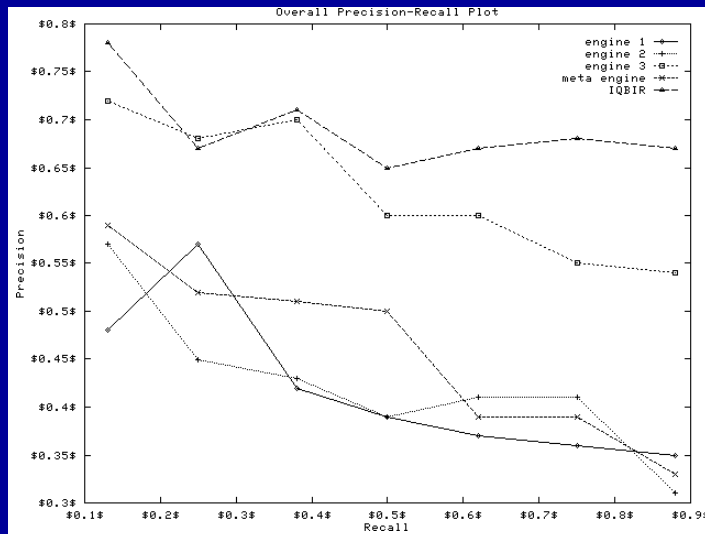
$$R = |A \cap R| / |A|$$

Expected search length [Cooper 68]

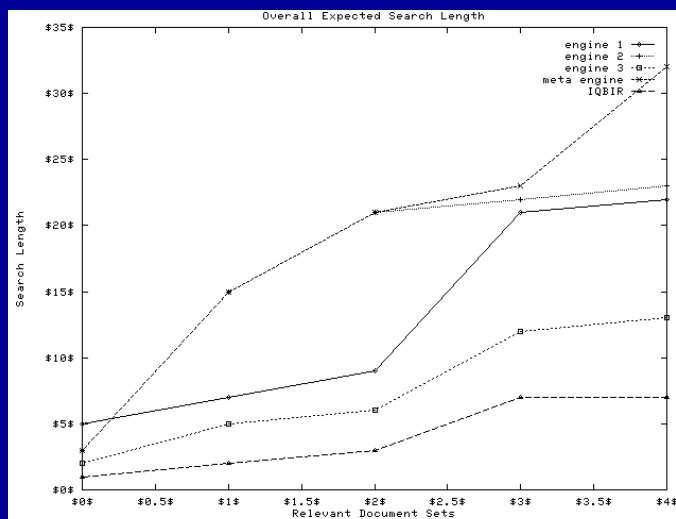
- N: desired number of most relevant documents
- Search length = number of irrelevant documents to skip through
- Example ranked list: 1 0 1 1 0 0 1 0 1 1 0
- Search length for goal 4 is 3.



Overall Precision Recall Results



Expected Search Length Results



Comments? Questions?

This presentation is posted at
<http://www.ece.wisc.edu/~hu/ibiqr.pdf>