

Seminars in Databases

Course Overview and Motivation

F.Ricci

Contact Details

- Francesco Ricci
 - Room 212 (POS)
 - fricci@unibz.it
 - 0471 016971
- Availability Hours: Wed 15.00-18.00
 - by prior arrangement via e-mail
- Course web site
 - <http://www.inf.unibz.it/~ricci/SDB/index.html>

Course Structure

- **Lectures: 24 hours**
- **Labs: 12 hours**
- **Timetable:**
 - Lectures: Wed 10.30-12.30
 - Labs: Wed 14.00-15.00
- **Assessment:**
 - Seminars: 70%
 - Final oral exam: 30%

Motivations

- Reading and UNDERSTANDING a scientific paper is not easy
 - 90% of the case you do not have all the background knowledge required to understand a paper
 - Understanding a paper is not a YES/NOT condition: you must decide when you have a "reasonable" understanding of the content
 - Most of the papers contains: mistakes, do not define all the important concepts, do not enter into the details of all the presented material.

Motivations

- Presenting a scientific paper is not easy
 - You must decide how to better allocate the various topics of the paper in the allowed time
 - You must decide the level of details to present for each section of the paper (normally not all the details can be presented)
 - You must (select) introduce additional material that is not contained in the paper (e.g. that in the references)
 - You must rise the attention of the audience
 - You must be able to adapt the presentation exploiting the (implicit) feedbacks from the audience

...Hence

- This course should enable you to practice and learn the difficult job of
 - reading a scientific paper
 - deciding when you have understood (90%) of the content
 - browsing/reading additional material that is going to help you to master the paper content
 - preparing an effective presentation
 - presenting complex ideas to somebody else
 - be able to derive your own conclusions, evaluations, further applications, from the presented ideas and techniques.

Course Format

- The teacher has selected a paper for each one of the 9 seminars (some more papers are listed in the web site)
- The teacher will illustrate the topic of each paper
- The students must aggregate in teams of 2 people
- The team can express their preferences over the papers
- The teacher will collect the preferences and will assign the papers to the team
- The number of papers assigned to each team could be different – *it is unavoidable*
- The teacher will schedule the seminars

Course Format

- The seminars will take place during the main lecture hours
- In the lab/exercise – the week before a seminar – the team that will do the next seminar can discuss with the lecturer:
 - the paper main issues
 - the structure of the presentation
- Do not expect that the teacher will “explain” the paper – the objective is to test if the students has caught the message contained in the paper and are almost ready to make the presentation.

What a student must do to pass

- ❑ Every student **must** have read the paper **when it is presented at the seminar**
- ❑ If something is not clear he/she must make a note and raise a question during the seminar
- ❑ The team should start reading the paper they are going to present two weeks before the seminar
 - To have enough time to fully understand the content
 - To be able to have a useful meeting with the lecturer the week before the seminar (in the LAB)
- ❑ During the seminar every student must raise relevant questions and comments
- ❑ In the discussion time – after paper presentation- every student must actively participate with comments (prepare them before the seminar!)

Exam

- ❑ The final grade is obtained evaluating the seminars and the knowledge acquired about the seminar topics in an oral final exam
- ❑ You will have two marks: S (for the seminars), and O for the oral part
- ❑ The final grade is $F = 0.7*S + 0.3*O$
- ❑ Both marks (O and S) must be greater or equal to 18 – you must pass both of them.

How seminars will be evaluated

- The presentation must follow the defined guidelines (see another slide)
- The presentation must cover (almost) all the topics contained in the paper – do not forget the important parts
- The presentation must be understandable and raise the audience attention
- The presenters must be able to reply to the questions of the other participants
- The presenters must demonstrate that they have understood the paper content.

How to make a seminar

- The presentation should take 1h (aprox)
- The presentation should contain around 40 slides
- The presentation should clearly describe:
 - The problems addressed in the paper (both from a technical and application point of view)
 - The approaches previously used to tackle the problem
 - A summary of the approach followed in the paper
 - The detailed description of the approach
 - The evaluation
 - The discussion and conclusion

How to make a seminar

- After the presentation of the paper the team, to activate the discussion, must make a short summary of the paper and formulate their opinion regarding:
 - The main techniques used in the paper
 - The problem addressed in the paper
 - If the problem has been solved, only partially, and what is still missing
 - The main advantages of the technique
 - The main disadvantages of the technique
 - How you believe you could further expand the work.

Topics

- Information retrieval and page ranking methods
- Structure of the web, hub and authorities
- Identification of Web communities
- Web search and geographical data
- To-k selection queries in relational databases
- Similarity based retrieval in metric spaces
- Database and k-means clustering
- Graph theory and recommender Systems
- Context-based personalization and data warehouse
- Personalization and query processing

Red line

- ❑ The Web has become a platform for service development and business innovation
- ❑ Software, such as browsers or communication clients, are becoming commodities, and value has moved to services and data
- ❑ The major Web players, such as Google or Yahoo, are companies that own and manage large databases (links or products) and can offer unique information services on top of that
- ❑ There is a need to develop advanced technologies for accessing large repositories of data typically generated in the web as a platform
- ❑ We shall investigate some of the most innovative techniques used nowadays to fully exploit various kinds of web data, such as links, multimedia objects, consumer generated content

Papers

1. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604-632, 1999.
2. Gary William Flake, Kostas Tsioutsoulouklis, and Leonid Zhukov. Methods for mining web communities: Bibliometric, spectral, and flow. In *Web Dynamics*, pages 45-68. 2004.
3. Taro Tezuka, Takeshi Kurashima, and Katsumi Tanaka. Toward tighter integration of web search with a geographic information system. In *15th International World Wide Web Conference WWW06*. 2006.
4. Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3-10, New York, NY, USA. ACM Press.
5. Bruno, N., Chaudhuri, S., and Gravano, L. (2002). Top-k selection queries over relational databases: Mapping strategies and performance evaluation. *ACM Trans. Database Syst.*, 27(2):153-187.
6. Charu C. Aggarwal, Joel L. Wolf, Kun-Lung Wu, and Philip S. Yu. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 201-212, New York, NY, USA, 1999. ACM Press.
7. Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103-145, 2005.
8. Gediminas Adomavicius, Alexander Tuzhilin, and Rhong Zheng. Rql: A query language for recommender systems. New York University, CeDER-05-15 2005.
9. Andreas Thor and Erhard Rahm. AWESOME - A data warehouse-based system for adaptive website recommendations. *Proceedings of the 30th VLDB Conference*, pages 384-395, Toronto, Canada, 2004.
10. Koutrika, G. and Ioannidis, Y. E. (2004). Personalization of queries in database systems. In *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004*, 30 March - 2 April 2004, Boston, MA, USA, pages 597-608.
11. Carlos Ordóñez. Programming the k-means clustering algorithm in sql. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 823-828, New York, NY, USA, 2004. ACM Press.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. J. ACM, 46(5):604-632, 1999.

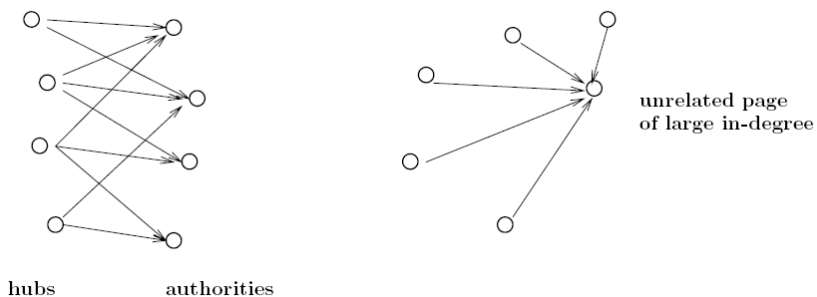
Problem

- Searching on the WWW – discovering pages that are relevant to a given query
- Hypothesis: The **network structure** of a hyperlinked environment can be a source of information **about the content** of the environment
- They consider three different types of queries
 - Specific: “what is the best search engine for the web”
 - Broad-topic: “find information about search engines”
 - Similar-pages: “find pages similar to” java.sun.com
- But they focus mostly on the second: the problem is to identify those pages that are more relevant/useful **among the large number of available options.**

Approach

- ❑ Develop a set of algorithms for extracting information from the link structures
- ❑ Hyperlinks encode a considerable amount of latent human judgment (*"if I link a page I think this is relevant and authoritative"*)
- ❑ Balance the "relevance" of the page (refer explicitly to the query) with the authority (has many links from other pages)
- ❑ The approach is based by observing the relationships between **authorities** about a topic and those pages that link to many related authorities (**hubs**)

Hubs and Authorities



Approach

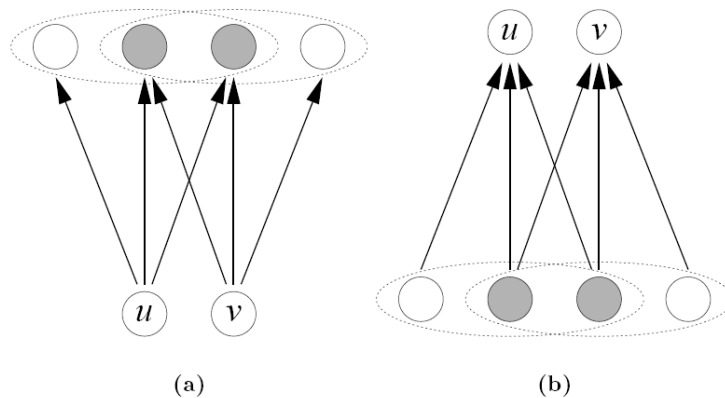
- ❑ First identify a set of pages that are related to a topic (using a text-based search engine) and are likely to contain the most authoritative pages for the searched topic
- ❑ Then, exploit the link structure of the pages “around” this initial set to identify both Hub and Authorities (HITS algorithm)
- ❑ Finally return the authorities of the given searched topic
- ❑ Techniques: graph theory, linear algebra (eigenvalues and eigenvectors)

Gary William Flake, Kostas Tsioutsoulis, and Leonid Zhukov. Methods for mining web communities: Bibliometric, spectral, and flow. In *Web Dynamics*, pages 45-68. 2004

Problem

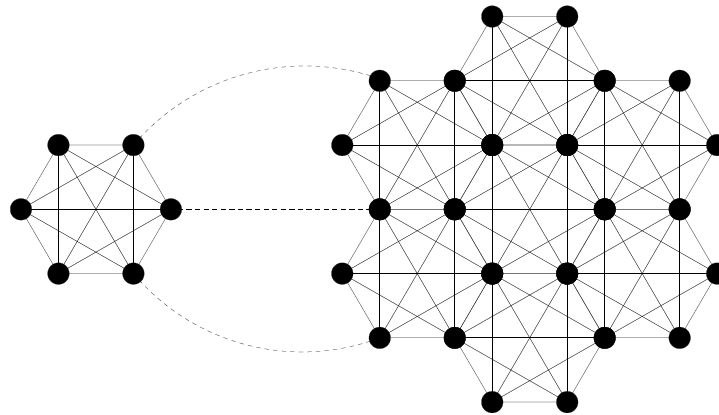
- A Web community is a collection of Web pages that are focused on a particular topic or theme (e.g. digital photography)
- The communities are pages that are more tightly coupled to each other than they are to pages outside the community
- It is a clustering problem where we want to group "similar" objects together and keeping "dissimilar" objects apart
- This problem differs from standard data mining: web is decentralized, evolving, enormous, contains a lot of noise.

Similarity



- a) **Bibliographic coupling:** the pages quoted by u and v overlap
- b) **Co-citation coupling:** The pages that quote u and v overlap

Example of two communities



Approach

- Communities are identified exploiting the implicit relations between documents formed by hyperlinks (not using the content)
- A community is defined as a subset of vertices with the property of having more edges connecting to other vertices in the community than to vertices not in the community
- They recast the community identification problem into a maximum flow framework
 - Given two vertices s and t in a graph, the s - t maximum flow problem is to find the maximum flow that can be routed from s to t while obeying all the capacity constraints

Experiments

□ Identify a set of Web sites dealing with the topic of September 11, 2001

- Howstuffworks "How Airport Security Works"
- Howstuffworks "How Biological and Chemical Warfare Works"
- Howstuffworks "How Black Boxes Work"
- Howstuffworks "How Building Implosions Work"
- Howstuffworks "How Cell Phones Work"
- Howstuffworks "How Cipro Works"
- Howstuffworks "How Cruise Missiles Work"
- Howstuffworks "How Emergency Rooms Work"
- Howstuffworks "How Machine Guns Work"
- Howstuffworks "How NATO Works"
- Howstuffworks "How Nostradamus Works"
- Howstuffworks "How Nuclear Bombs Work"
- Howstuffworks "How Skyscrapers Work"
- Howstuffworks "How Stun Guns Work"
- Howstuffworks "How the U.S. Draft Works"
- Howstuffworks "How Viruses Work"

Members of the 9/11 community contained in the Howstuffworks website

T. Tezuka, T. Kurashima, and K. Tanaka.
Toward tighter integration of web search with a
geographic information system. In 15th
International World Wide Web Conference
WWW06. 2006

Problems

- ❑ There are already a number of local Web search systems enabling user to find location specific web content (google maps)
- ❑ There is still much to do
 - If your are driving you cannot pay much attention to a map interface
 - If the list of retrieved item is too long it could take too long to open all pages and check the content
 - A user may be interested in learning people's opinions of a region (what is said in the web).

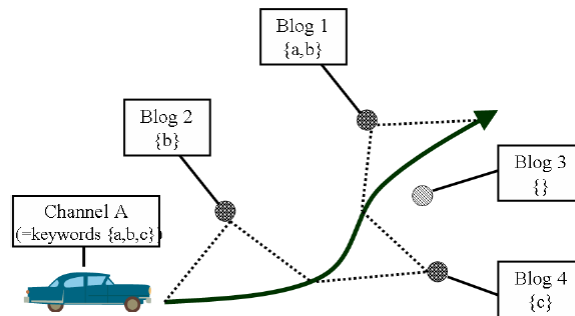
Google Maps search results for "ottica" near Bolzano, Italy. The search results list several businesses with their addresses and phone numbers. A map on the right shows the location of these businesses in Bolzano, Italy, with red pins labeled A through J.

Results: 1-10 of about 270 for **ottica** near 39100 Bolzano, Italy

Categories: [Ottica - vendita al dettaglio](#)

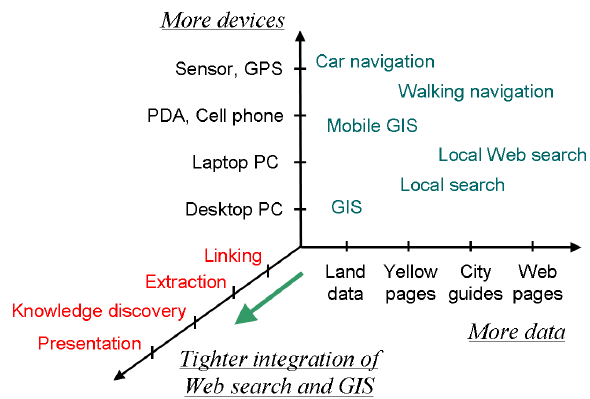
- A** [Ottica Matt S.R.L.](#)
Piazza Von Der Vogelweide Walther, 12,
39100 Bolzano (BZ), Italy
0.3 km W - 0471 301314
- B** [Schrott Sas](#)
Via Dei Bottai, 32, 39100 Bolzano (BZ), Italy
0.3 km N - 0471 970695
- C** [Walter Ottica S.R.L.](#)
Via Dei Portici, 13, 39100 Bolzano (BZ), Italy
0.4 km NW - 0471 973522
- D** [Aerre S.n.c.](#)
Piazza Domenicani, 27, 39100 Bolzano (BZ), Italy
0.5 km W - 0471 973749
- E** [Wassermann](#)
Via Dei Portici, 72/B, 39100 Bolzano (BZ), Italy
0.5 km NW - 0471 980677
- F** [Walter Ottica S.R.L.](#)
Via Leonardo Da Vinci, 4, 39100 Bolzano (BZ), Italy
0.6 km W - 0471 973336
- G** [Optik Leitner](#)
Via Museo, 8, 39100 Bolzano (BZ), Italy

Web car radio



Approach

- A tighter integration of web search with geographic information systems (GIS)




Extraction

- Extract from a local web search only those that contain regional information
- They extract from blogs experiences related to specific places and times
- They use this information extracted from blogs to compile "rules"
 - E.g.: in spring visit that famous sightseeing spots to enjoy flower blossoms
- The goal is to present tourists' real-life experiences through a map interface.

Knowledge Discovery

- Aggregating extracted information into knowledge
- Examples
 - Summarizing the content into an overview
 - Removing overlaps among content
 - Identifying rules or patterns in the content
 - Identifying statistical tendencies in the content data
- They focus on the discovery of important place names (landmark) and in ranking them by their significance based on the way they appear on the Web content.

Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006). Learning user interaction models for predicting web search result preferences. In SIGIR '06, pages 3-10, New York, NY, USA. ACM Press



Problem

- ❑ Relevance measurement is crucial to web search
- ❑ Traditionally, relevance is measured using human assessors to judge relevance of query-document pairs
- ❑ BUT this is expensive
- ❑ Why not using the “implicit” feedback provided by people interacting with search engines?

Approach

- Implicit feedback
 - If a user clicks on a link (in a search result) and not in the previous one this indicates that that link is more relevant than the other
 - If a user visits a link and read (spend time on that) the page content this indicates that the page is relevant
- Sometime these conclusions are not reliable: irrational behavior, malicious users, or fake users
- BUT Observing the aggregate behavior of large number of user one can correct the noise inherent in individual interactions

Example

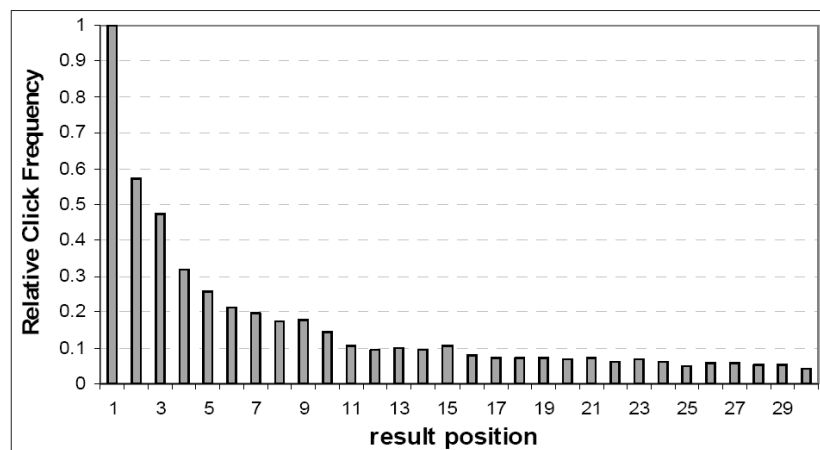


Figure 3.1: Relative click frequency for top 30 result positions over 3,500 queries and 120,000 searches.

Correcting click frequency for relevance estimation

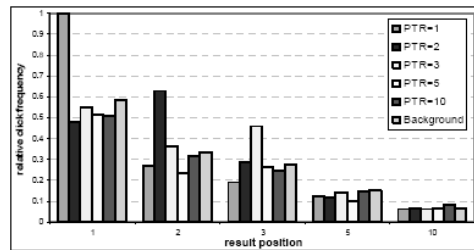
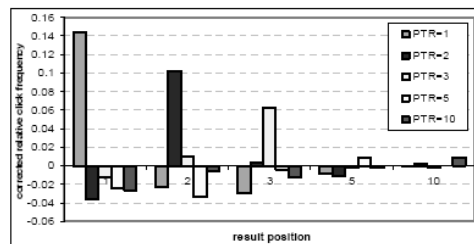



Figure 3.2: Relative click frequency for queries with varying PTR (Position of Top Relevant document).



Approach

- Identify important features for modeling post search interaction (e.g. time spent on page, click frequency)
- Train a predictive behavior model that given the features predict the relevance of the link (on a set of queries for which we know the relevance of the results)
- Use this model to predict the relevance of the links (or preferences among links) on a set of queries for which we do not know the best ranking of links

Bruno, N., Chaudhuri, S., and Gravano, L. (2002). Top-k selection queries over relational databases: Mapping strategies and performance evaluation. ACM Trans. Database Syst., 27(2):153-187



Problem

- Approximate matches of queries
- "I want a house with a certain price and number of bedrooms"
- The database system should rank the available houses according to how well they match the given user preferences and return the top houses
- If there is no exact match the system must return first those more "similar".

Property Example

Case	Location code	Bedrooms	Recep rooms	Type	floors	Condition	Price
1	8	2	1	terraced	1	poor	20,500
2	8	2	1	terraced	1	fair	25,000
3	5	1	2	semi	2	good	48,000
4	5	1	2	terraced	2	good	41,000

Probe case = query

Case	Location code	Bedrooms	Recep rooms	Type	floors	Condition	Price
5	8	2	2	semi	2	fair	20,500

 = matching attribute

Problem

- ❑ These queries are not currently supported (the indexes and the access methods) by many RDBMS
- ❑ Avoid full sequential scan of the data
- ❑ Provide this functionality for a wide variety of ranking functions (distance metrics)
- ❑ Distance Metrics:

$$Sum(q, t) = \|q - t\|_1 = \sum_{i=1}^n |q_i - t_i|$$

$$Eucl(q, t) = \|q - t\|_2 = \sqrt{\sum_{i=1}^n (q_i - t_i)^2}$$

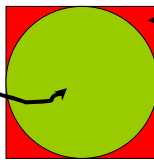
$$Max(q, t) = \|q - t\|_\infty = \max_{i=1}^n |q_i - t_i|$$

Approach


- ❑ **Not** to develop a new stand-alone algorithm and data structure for the nearest-neighbor problem over multidimensional data
- ❑ Mapping a top- k selection query to a traditional range selection query that can be optimized and executed by any RDBMS
- ❑ The key issue is to determine (given the query and the data) the right size of the ranges in the range query
 - If too large then too many records must be compared with the query
 - If too small then we can select not enough records to compare

Approach

- Search** Given a top- k query q over R , use a multidimensional histogram H to estimate a search distance d_q , such that the region $reg(q, d_q)$ that contains all possible tuples at distance d_q or lower from q is expected to include k tuples (Section 3.1).
- Retrieve** Retrieve all tuples in $reg(q, d_q)$ using a range query that encloses this region as tightly as possible (Section 3.2).
- Verify/Restart** If there are at least k tuples in $reg(q, d_q)$, return the k tuples with the lowest distances. Otherwise, choose a higher value for d_q and *restart* the procedure (Section 3.3).



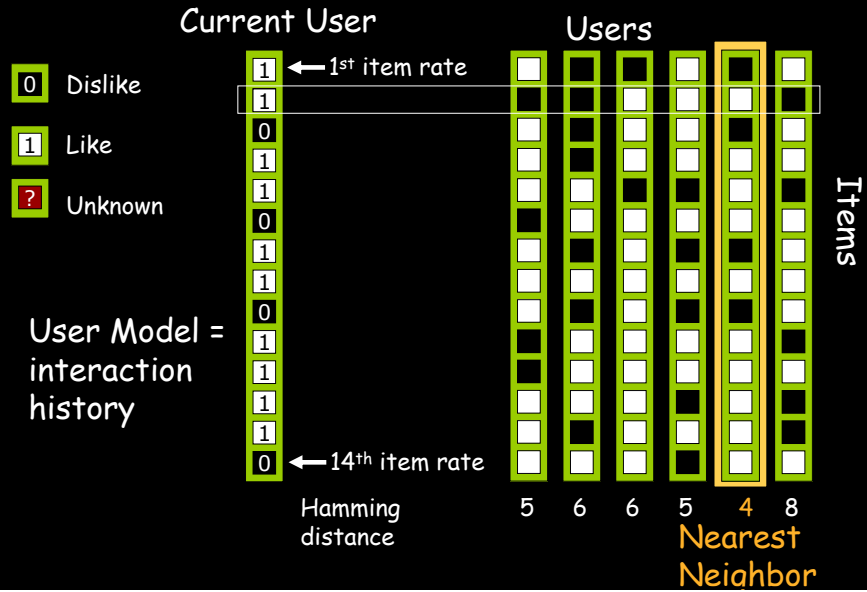
Charu C. Aggarwal, Joel L. Wolf, Kun-Lung Wu,
and Philip S. Yu. Horting hatches an egg:
a new graph-theoretic approach to
collaborative filtering. In KDD '99, pages
201-212, New York, NY, USA, 1999. ACM
Press



Problem

- ▣ In recommender systems based on collaborative filtering users are recommended items that liked to similar users
- ▣ A major problem is related to the fact that is difficult to have enough data (ratings of items) to measure in a reliable way the user similarity.

Nearest Neighbor Collaborative-Based Filtering



Matrix of ratings

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a				1	4	5			4	3						2			4	2					
b			4							3							5	1		3					
c		5		4			4				3		5						4		4		5		
d								3			5						3			4	2				3
e		3					5			4	5				5					1			5		4
f			4				1	3	5		4	1		5	4	4			4					3	
g	2	4				4	2			5		1	4	5	4	2	4			5				4	
h			2	1		4		3	5		4	2		5	4	5								5	
i			1				3			5			5		4	4			5			4		3	
j		4				4				5		1	5		5	4	4			4			4		
k		5				4			2	5		1	5		4		2		4					2	
l						3			3		4	1		4	4	2	4							3	
m	5	3					5	3		5	4		5	5	3				4	4	5	4		4	
n			1		4	5				4	5		1	5		4		3		4		4	3		
o		4				4				5	4			5			4	2		5		5		3	
p			4				5						5	4		5	4		2	4	4	5	4		2
q				3				3				1	5		4	4		4		4			4		3
r		4		1	4			2				2	5		4						5	4		4	
s		2	4			4			5			1		4		4	2	4		4		4		5	
t		1		4			3				4		5	5		4		4		4			4		3
u		2		1		4		3				1	5	4		2	4		5	4					
v				4	5					4	3		5		2					2		2		5	
w			2			2	3			3		5			4	5		4	2		3	4			
x	4			5			3			3			4	5						1					
y			1				3				2	3					3	3		5		4			

Approach

- They introduce the notions of **horting** and **predictability**
- A user A horts a user B if B has rated the “majority” of items rated by A
- A user B predicts a user A iff A horst B and there is linear transformation that maps (reasonably well) the rating of B into the ratings of A
- They user a graph theoretic approach where the horting and predict relations are the labels of a graph having the users as vertices.

Results

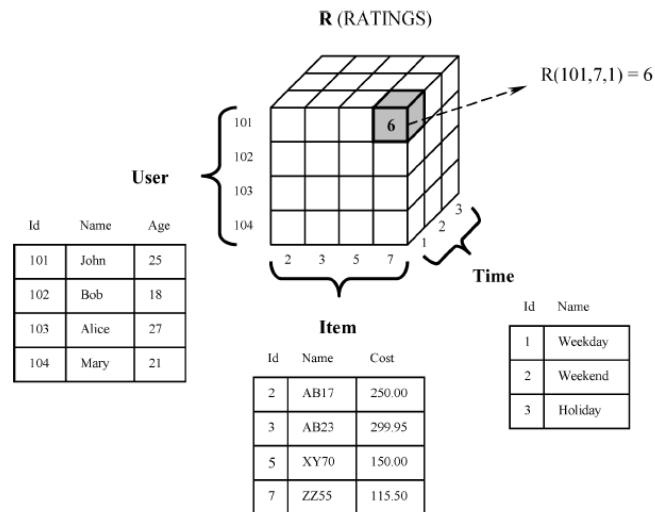
- They compare their approach to other (simpler) ones and show that they can better predict the ratings of a user

G. Adomavicius, R. Sankaranarayanan, S. Sen,
and A. Tuzhilin. Incorporating contextual
information in recommender systems using a
multidimensional approach. ACM Trans. Inf.
Syst., 23(1):103-145, 2005

Bi-dimensional vs. multidimensional

- The standard model for a recommender system is $(R: \text{Users} \times \text{Items} \rightarrow [0,1] \cup \{?\})$ **bi-dimensional**
- A general model may include some “contextual” dimensions, e.g.:
 - $R: \text{Users} \times \text{Time} \times \text{Place} \times \text{Items} \rightarrow [0,1] \cup \{?\}$
- This is the approach of a multidimensional data model developed for datawarehousing and OLAP

Multidimensional Model



Recommendation Problem

- Assume that the rating function is complete (defined for each entry in $D_1 \times D_2 \times \dots \times D_n$)
- Recommendation problem:
 - "what" to recommend is a subset of the dimensions: D_{i_1}, \dots, D_{i_k} ($k < n$)
 - "for whom" is another subset of the dimensions: D_{j_1}, \dots, D_{j_l} ($l < n$)
 - The dimension in "what" and "for whom" have a void intersection, and

$$\begin{aligned}
 & \forall (d_{j_1}, \dots, d_{j_l}) \in D_{j_1} \times \dots \times D_{j_l}, \quad (d_{i_1}, \dots, d_{i_k}) = \\
 & \quad \arg \max_{\substack{(d'_{i_1}, \dots, d'_{i_k}) \in D_{i_1} \times \dots \times D_{i_k} \\ (d'_{j_1}, \dots, d'_{j_l}) = (d_{j_1}, \dots, d_{j_l})}} R(d'_1, \dots, d'_n)
 \end{aligned}$$

Approach


- They try to understand when the context is relevant or not: search for the correct segments
- When the context is relevant they use it to build a prediction
- When not relevant they use a traditional approach: merge the ratings obtained in different contexts and use a bi-dimensional approach

Summary of the differences

Segment	Segment-based F-measure	Whole-data-based F-measure
Theater-Weekend	0.641	0.528
Theater	0.608	0.479
Theater-Friends	0.607	0.504
Weekend	0.542	0.484

- Substantial improvement of F-measure on some segments
- Since Theater-Friends has lower F-measure than Theater then this is discarded (see the original algorithm)
- The final segments obtained are: Theater-weekend, theater and weekend.

Gediminas Adomavicius, Alexander Tuzhilin, and Rhong Zheng. Rql: A query language for recommender systems. New York University, CeDER-05-15 2005



Problem

- ❑ As in the previous paper the goal is to be able to make recommendations in different contexts
- ❑ Classical recommender can only recommend “good” items to the target user
- ❑ There are other “queries” related to the recommendation problem that could be interesting to solve

Examples

- ❑ Recommend the best movies to users
- ❑ Recommend, using personal ratings, top 5 action movies to users older than 18
- ❑ Recommend top 5 movies to the user to see over the weekend, but only if the personal ratings of the movies are higher than 7
- ❑ Recommend to Tom and his girlfriend top 3 movies and their best times to see them over the weekend
- ❑ Recommend movie genre to different professions using only the movies with personal ratings bigger than 6
- ❑ Identify the top two professions that appreciate the movie "Beautiful Mind" the most

Approach


- ❑ They define a Recommendation Query Language

```
RECOMMEND recommend_dim_attr_list
TO recipient_dim_attr_list
FROM cube
BASED ON measure_list
WHERE dimension_restrictions //optional
WITH measure_restrictions //optional
AGGR BY aggregation_dim_attr_list //optional
HAVING aggregation_restriction //optional
SHOW measure_rank_restriction
//optional, default: SHOW TOP 1
```

Approach

- They define a recommendation algebra
- They show how to translate from the recommendation query language to the recommendation algebra
- They translate from the recommendation algebra to the relational algebra
- An finally from the relational algebra to SQL

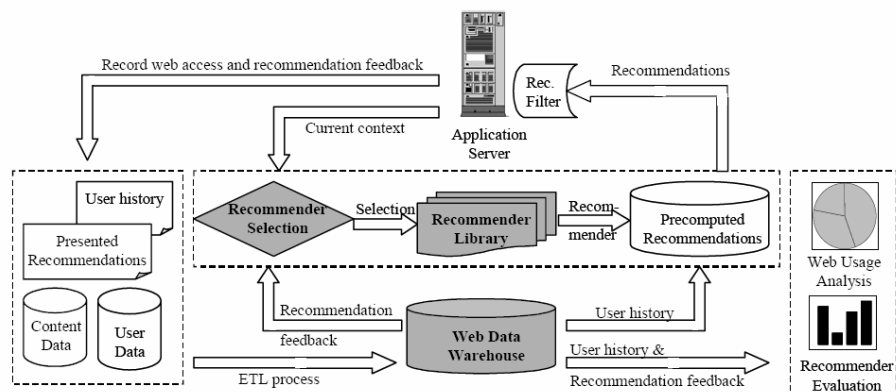
Andreas Thor and Erhard Rahm. AWESOME -
A data warehouse-based system for adaptive
website recommendations. Proceedings of the
30th VLDB Conference, pages 384-395,
Toronto, Canada, 2004



Problem

- Recommendations can be generated in a number of ways and exploiting a range of information
 - Using product characteristics, user characteristics, buying history
 - Recommend top selling products, new products, products similar to those bought in the past, products bought by similar users, products bought together with other products currently considered, ...
- BUT little information is available on the relative quality of different recommenders
- GOAL: comparative qualitative evaluation of different recommenders.

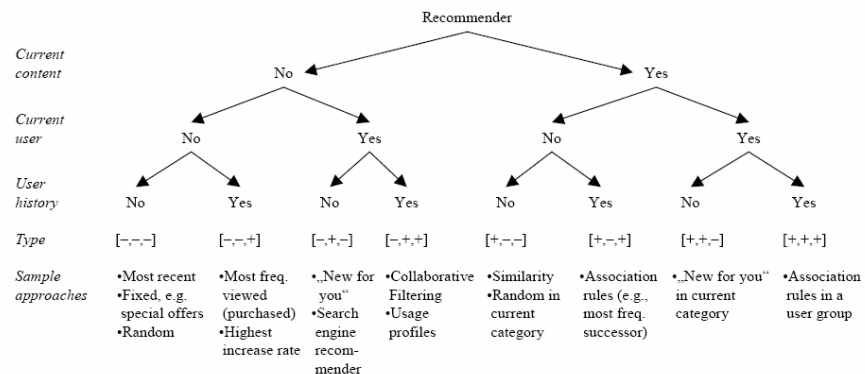
AWESOME Architecture



Awesome

- ❑ Dynamically select the most promising recommender system
- ❑ Then selecting the relevant recommendations for the given context
- ❑ Selection is based on the measure of the recommendation quality
- ❑ Awesome can “adapt” to the changes in the user population and content
- ❑ It is based on a data warehouse approach
 - Website structure, content, website users and customers, website usage history and recommendation feedback.

Classification of recommender



Evaluation of recommenders


- Accepted, Viewed and Purchased measures
 - **Acceptance rate** = the percentage of page views for which at least one presented recommendation was accepted
 - **Session acceptance rate** = the percentage of sessions for which at least one presented recommendation was accepted
 - **View rate** = the percentage of page views for which any of the presented recommendation was reached later in the session
 - **Session view rate** = percentage of the sessions with at least one page views presented a recommendation that was reached later

Acceptance rates

Recommender		User type		
Type	Name	New users	Returning users	Σ
[-,-,-]	Most recent	(0.42%)	(0.00%)	(0.38%)
[-,-,+]	Most frequent	1.00%	0.62%	0.92%
[-,+,-]	SER	2.84%	1.95%	2.79%
[-,+,+]	Personal Interests	–	1.54%	1.54%
[+,-,-]	Similarity	1.65%	0.82%	1.56%
[+,-,+]	Association Rules	1.16%	0.68%	1.08%
	Σ	1.82%	1.09%	1.69%

SER is used when the user arrives to the site through a search engine. SER recommends the products that are not in the page shown but matches the keywords entered in the search engine.

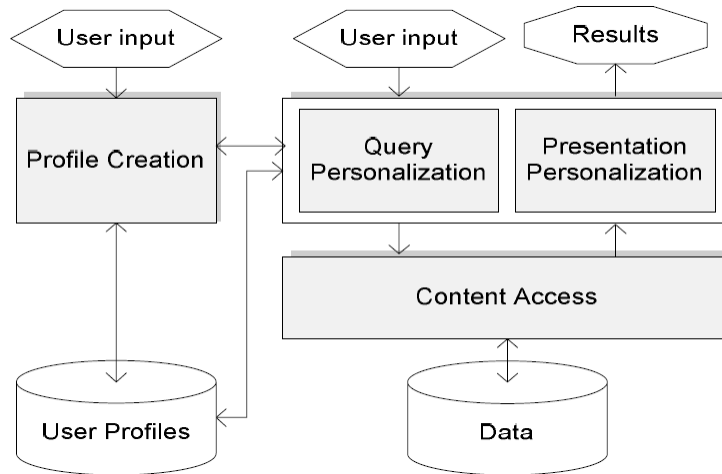
Koutrika, G. and Ioannidis, Y. E.
Personalization of queries in database systems.
In Proceedings of ICDE 2004, 30 March - 2
April 2004, Boston, MA, USA, pages 597-608



Problems

- ❑ Given a query to a database the result is not going to change if the user changes (right!)
- ❑ Consider a case where users Julie and Rob both inquiring about what is shown tonight
 - Julie likes comedies and thrillers
 - Rob likes sci-fi and actress J. Roberts
- ❑ User preferences can be stored in a user profile
- ❑ The system could integrate the user preferences and offer to Julie and Rob two different result lists – each one ranked according to the user preferences.

Approach



Approach

- ▣ The user preferences are stored as atomic selection or join conditions
- ▣ Preferences are expressed in the form of degree of interest (a number in $[0,1]$)

```
[ THEATRE.tid=PLAY.tid,      1 ]  
[ PLAY.tid=THEATRE.tid,    1 ]  
[ PLAY.mid=MOVIE.mid,      1 ]  
[ MOVIE.mid=PLAY.mid,      0.8 ]  
[ MOVIE.mid=GENRE.mid,     0.9 ]  
[ ACTOR.name='A. Hopkins',  0.8 ]  
[ GENRE.genre='comedy',     0.9 ]  
[ GENRE.genre='thriller',   0.7 ]
```


Approach

- User preferences that are “adjacent” can be combined to build transitive user preferences
 - For instance if there is a preference on W.Allen (0.7) and a general interests for the director of the movie (0.8) then the preference for W.Allen as director is $0.7 * 0.8$
- User preferences can be combined (conjunction and disjunction) hence a formula for these combination is given.

Approach

- Given a query and a user profile the system must build a new query that integrates the original query and the “relevant” preferences:
 - Step 1: select the relevant preferences from the user profile
 - Step 2: integrate the selected preferences in the query and generate one or more new queries
- The paper compare two different methods for integrating preferences into the query (single query and multiple queries).

Carlos Ordonez. Programming the k-means clustering algorithm in sql. In KDD '04, pages 823-828, New York, NY, USA, 2004. ACM Press



Problem

- ❑ There are many useful clustering algorithms in DM literature
- ❑ Typically you implement them by minimizing disk access and doing most of the work in main memory
- ❑ Hence this is difficult to be done on a real LARGE database
- ❑ Can SQL be used to get an efficient implementation of a clustering algorithm?

K-means

- Works when we know k , the number of clusters we want to find
- Idea:
 - Randomly pick k points as the “centroids” of the k clusters
 - Loop:
 - For each point, put the point in the cluster to whose centroid it is closest
 - Recompute the cluster centroids
 - Repeat loop (until there is no change in clusters between two consecutive iterations.)

Approach

- Define a set of additional tables to store temporary data: e.g. distances between points
- Defining appropriate SQL queries for:
 - Initializing K-means
 - Computing the Euclidean distances
 - Finding the nearest centroid
 - Updating the clustering results
- Find methods to optimize the computation

End

▣ Now is your turn ...