

A personalised information support system for searching portals and e-resources

B. S. Sirisha, V. K. J. Jeevan, R. V. Raja Kumar, A. Goswami

Authors:

B.S. Sirisha was Project Assistant and MS Student, Indian Institute of Technology, Kharagpur, E-mail: b_s_sirisha@yahoo.com; V.K.J. Jeevan is Deputy Librarian, Library and Documentation Division, Indira Gandhi National Open University, Maidan Garhi, New Delhi, E-mail: vkj@rediffmail.com; R.V. Rja Kumar is Professor, Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, E-mail: rkumar@ece.iitkgp.ernet.in; and A. Goswami is Professor, Department of Mathematics, Indian Institute of Technology, Kharagpur, E-mail: goswami@maths.iitkgp.ernet.in

Structured Abstract

Research paper

Purpose

To describe the development of a personalised information support system to help faculty members to search various portals and e-resources without typing the search terms in different interfaces and to obtain results re-ordered without human intervention.

Design/methodology/approach

After a careful survey of various tools and techniques available for computerised client-centred information services, the study selected to apply web usage mining, proxy level data collection and a vector space retrieval model to develop the personalised information support for teaching and research in a higher education institution.

Findings

The paper reports the preliminary results of search term generation to match the subject interests of faculty members using clustering.

Research limitations/implications

The paper has not considered cases for all the faculty members due to time constraints. The results obtained from the system also need correlation with the sources actually used by the faculty to test its efficacy in a highly fluid research interest situation like higher academics.

Practical implications

A pragmatic client-centred information support prototype described in this study may find use in other institutions needing similar information support.

Originality/value

This paper demonstrates the pragmatic application of ICT for linking users and e-resources in an academic library.

Keywords: Personalisation; India; Higher education institutions; Web usage mining; Proxy-based data collection methods; Vector space modelling; IIT Kharagpur

Word length: 6862

1. Introduction

The number of pages on the Web is growing at an exponential rate. As a result of this, search engines may ignore a user's preferences during the search process, and may return a large amount of irrelevant data. One way to prevent this is by personalising search results through careful mapping of a user's preference. Personalisation aims at achieving relevant search results through automatic means by self-learning from the user's interaction with the retrieval system. It chooses content for the user automatically, without direct user request, and as the system becomes more familiar with user habits by observing behaviour, it achieves increased accuracy in predicting future behaviour and interests

(http://download.oracle.com/docs/cd/A97329_03/web.902/a95883/personal.htm). Kramer et al. (2000) suggest personalisation as “a toolbox of technologies and application features; from simple display of the end-user's name on a web page, to complex catalog navigation and product customization based on deep models of users' needs and behaviors”. User satisfaction is the ultimate aim of personalization, motivated by the recognition that a user has needs and meeting them successfully is likely to lead to a satisfying relationship and enhanced use of the services offered. Beyond the common goal, however, there is great diversity in how personalisation can be achieved. Information about the user can be obtained from a history of previous sessions, or through interaction in real time.

Different systems have been developed to assist users with web browsing based on users' feedback, by explicitly asking them for page recommendations. The majority of the systems that display personalisation characteristics belong to one of the following categories (Eirinaki and Vazirgiannis, 2003):

- *Manual decision rule systems* - these are framed by the creator based on static data, such as user demographics acquired through a registration process, or dynamic data, such as session histories (Vozalis et al., 2001). Systems that belong to this category are Yahoo's personalisation engine (Manber et al., 2000) and Broadvision (<http://www.broadvision.com>).
- *Content-based systems* - these rely on content similarity between web documents and personal profiles. The system takes some user interests as an initial input and

then updates these interests based on the pages visited. The system then recommends pages based on these interests and the previous browsing behaviour of other users with similar interests (Vozalis et al., 2001). A system that fits in this category is WebWatcher (Joachims et al, 1997).

- *Collaborative filtering systems* - these content-based collaborative filtering systems take explicit information in the form of user ratings or preferences and, through a correlation engine, return information that is predicted to closely match the users' preferences (Vozalis et al., 2001). Systems like Netperceptions (<http://www.netperceptions.com>) incorporate collaborative filtering techniques. The collaborative filtering doesn't scale well and applying this technique to a large number of items can cause prediction performance and accuracy to suffer since many items aren't rated (Anand and Mobasher, 2005).
- *Web usage mining systems* - since the first three of the above systems rely on user supplied personal data which is prone to bias, and static profiles degrade the system too over time (Anand and Mobasher, 2005), auto tracking of user interests is attempted in web usage mining systems. These systems attempt to incorporate techniques for pattern discovery from web usage data with the help of a number of web mining algorithms (Vozalis et al., 2001). Bluemartini (<http://www.bluemartini.com>) and Webtrends (<http://www.webtrends.com>) come under this category.

This paper presents the results of systems under development in the Indian Institute of Technology (IIT), Kharagpur. The systems aim to search automatically and retrieve information resources of interest from online journals and databases for the faculty members of IIT, Kharagpur who are actively engaged in the frontier areas of science and technology teaching and research using web usage mining, proxy data, and vector space modelling. Jeevan and Padhi (2006a) review the literature on content personalisation and Jeevan and Padhi (2006b) report on the preparedness of other libraries in IITs for this development.

2. Web usage mining

Web usage mining can reduce the need for obtaining subjective user ratings or registration based personal preferences. Lieberman (1995) suggests learning the interests of a user by observing their browsing behaviour and then recommending which links to follow. It models the browsing process, rather than explicitly modelling the user. WebWatcher (Joachims et al., 1997) takes some user interests as an initial input and updates these interests based on pages then visited. The system then recommends pages, based on these interests and the previous browsing behaviour of other users with similar interests. WebMate (Chen and Sycara, 1998) is an agent that assists browsing and searching using multiple term vectors for different domains of user interest and updating these incrementally when users give positive feedback for visited pages. WebACE (Han et al., 1998) constructs a customised user profile by recording information about the documents the user browses. It then clusters these documents, using novel clustering techniques, and uses these to generate queries to search for similar documents. Personal View Agent (Chen et al., 2001) tracks, learns and manages user interests. Beginning with a fixed palette of categories, the system follows the user, detecting their domains of interest. This 'personal view' takes the form of a tree and can adapt to changing user interests using a 'personal view maintainer', which can split and merge categories in the personal view.

Web usage mining helps to track automatically user interests out of their initial interaction with the system. The architecture for web usage mining consists of the following two components (Zhang and Chang, 2002):

- data pre-processing
- data mining.

2.1 Data pre-processing

The steps involved in pre-processing (Cooley et al., 1999) of the server log are as follows

- a. Data cleaning.* Since the server log contains irrelevant information such as background images, it has to be cleaned first. Records about image files (.gif, .jpg, .tif) are filtered as well as unsuccessful requests (Diebold and Kaufmann, 2001).

- b. *Session and user identification.* Requests from the same IP address are grouped into a session. A timeout of 30 minutes is used to decide the end of a session, i.e., if the same IP address does not occur within a time range of 30 minutes, the current session is closed. Subsequent requests from the same IP address will be treated as a new session. The time spent on a particular page is determined by the time difference between two consecutive requests. The server log files are transformed into a set of sessions. A session represents a single visit of a user. There are some difficulties in accurately identifying sessions and estimating times spent on pages, due to client or proxy caching of pages, IP sharing, network congestions, and interruptions.
- c. *Path completion.* This refers to the problem of inferring missing user references due to caching. Effective path completion requires extensive knowledge of the link structure within the site. Referrer information in server logs can also be used in disambiguating the inferred paths.
- d. *Formatting.* Once the appropriate pre-processing steps have been applied to the server log, a final preparation module can be used to properly format the sessions or transactions for the type of data mining to be accomplished. For example, since temporal information is not needed for the mining of association rules, a final association rule preparation module would strip out the time for each reference and do any other formatting of the data necessary for the specific data mining algorithm to be used.

2.2 Data mining

This is the process of extracting data patterns from a large amount of data. The processes (Srivastava et al., 2000) involved are:

- a. *Log file analysis.* This is probably the most widely used technique to obtain structured information out of server logs. It will provide information such as the number of hits and page views, the number of unique and returning users, the average length of a page view, an overview of the browsers and operating systems that were used, and an overview of keywords that were used in search engines and that led to the website.

- b. Association rules.* Association rules (Agrawal and Srikant, 1994) mining is a technique for finding frequent patterns, associations, and correlations among sets of items. Association rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests.
- c. Sequential patterns.* Sequential pattern (Cooley et al., 1997) discovery is an extension of association rules mining in that it reveals patterns of co-occurrence incorporating the notion of time sequence. In the web domain such a pattern might be a web page or a set of pages accessed immediately after another set of pages. Using this approach, useful users' trends can be discovered, and predictions concerning visit patterns can be made.
- d. Clustering.* This is used to group together items that have similar characteristics. In the context of web mining, two cases can be distinguished, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users' perception. User clustering results in groups of users that seem to behave similarly when navigating through a website (Steinbach, 2000).
- e. Classification.* Classification (Han et al., 1993) is the task of mapping a data item into one of several predefined classes. Classification can be done by using supervised inductive learning algorithms such as decision tree classifiers, naive Bayesian classifiers, k-nearest neighbour classifiers, and Support Vector Machines.

3. Proxy level data collection

The data for the web usage mining is obtained in three ways:

- a. Server level collection:* explicitly records the browsing behaviour of site visitors. The data recorded in server logs reflects the (possibly concurrent) access of a website by multiple users. These log files can be stored in various formats such as 'common log' (<http://www.w3.org/Daemon/User/Config/>) or 'extended log'

(<http://www.w3.org/TR/WD-logfile.html>) formats. However, the site usage data recorded by server logs may not be entirely reliable due to the presence of various levels of caching within the web environment

(http://www.arena.no/nedlasting/dokumentasjon/wt_smartsources_r3.pdf). Cached page views are not recorded in a server log. In addition, any important information passed through the 'POST' method (a command used in html forms to store or update data) will not be available in a server log. Packet 'sniffing' technology is an alternative method to collect usage data through server logs. Packet sniffers are rarely used in practice because of the scalability issue on web servers with high traffic. The web server can also store other kinds of usage information such as cookies and query data in separate logs. Cookies are tokens generated by the web server for individual client browsers in order to track automatically the site visitors (Srivastava et al., 2000). Tracking of individual users is not an easy task due to the stateless connection model of the HTTP protocol.

b. Client level collection: can be implemented by using a remote agent, such as Java scripts or Java applets (Shahabi and Banaei-Kashani, 2002) or by modifying the source code of an existing browser, such as Internet Explorer or Mozilla (Catledge and Pitkow, 1995) to enhance its data collection capabilities. The implementation of client-side data collection methods requires user co-operation, either in enabling the functionality of the Java scripts and Java applets, or voluntarily to use the modified browser. Client-side collection has an advantage over server-side collection because it ameliorates both the caching and session identification problems. However, Java applets perform no better than server logs in terms of determining the actual view time of a page. In fact, it may incur some additional overhead especially when the Java applet is loaded for the first time. Java scripts, on the other hand, consume little interpretation time but cannot capture all user clicks (such as reload or back buttons). These methods will collect only single-user, single-site browsing behaviour. A modified browser is much more versatile and will allow data collection about a single user over multiple websites. The most difficult part of using this method is convincing the users to use the browser for their daily browsing activities (Srivastava et al., 2000).

c. Proxy level collection: unlike individual web servers that contain only a limited number of web pages, a proxy server, sitting in the middle-tier in the Internet

infrastructure, serves many web clients and covers a wide scale of the web domain consisted of heterogeneous websites. Proxy caching can be used to reduce the loading time of a web page experienced by users as well as the network traffic load at the server and client sides. The performance of proxy caches depends on their ability to predict future page requests correctly. Proxy traces may reveal the actual HTTP requests from multiple clients to multiple web servers. This may serve as a data source for characterising the browsing behaviour of a group of anonymous users, sharing a common proxy server (Srivastava et al., 2000).

4. Vector space model

In order to run mining algorithms, the web data obtained from the proxy server has to be converted into a quantifiable format using one of the following three methods (Belkin and Croft, 1992):

a. Boolean Model: this is based on the concept of an exact match of a query specification with one or more text surrogates. Unfortunately this model is constrained by the following:

- Its retrieval strategy is based on a binary decision criterion (i.e. a document is predicted to be either relevant or not relevant) without any notion of a grading scale, which prevents good retrieval performance.
- While Boolean expressions have precise semantics, frequently it is not simple to translate an information need into a Boolean expression.

b. Probabilistic Model: the probabilistic model was introduced by Robertson and Sparck Jones (1976) and was later known as the binary independence retrieval (BIR) model. The probabilistic model attempts to capture the information retrieval problem within a probabilistic framework. The fundamental idea is: given a user query, there is a set of documents which contains exactly the relevant documents and no other. This set of documents is referred as the ideal answer set. Giving the description of this ideal answer set, we would have no problems in retrieving its documents. Thus, we can think of the querying process as a process of specifying the properties of an ideal answer set. The

main advantage of the probabilistic model is that documents are ranked in decreasing order to their probability of being relevant. However, the limitations include:

- The need to guess the initial separation of documents into relevant and non-relevant sets.
- The fact that the method does not take into account the frequency with which an index term occurs inside a document (i.e., all weights are binary).
- The adoption of the independence assumption for index terms.

c. Vector space model: this model improves the Boolean model by assigning non-binary weights to index terms in queries and in documents. These term weights are ultimately used to compute the degree of similarity between each document stored in the system and the user query. By sorting the retrieved documents in decreasing order of this degree of similarity, the vector space model takes into consideration documents that match the query terms only partially. The consequence is that the ranked documents answer set is a lot more precise than the document answer set retrieved by the Boolean model.

The main advantages of the vector space model are:

- Its term-weighting scheme improves retrieval performance.
- Its partial matching strategy allows retrieval of documents that approximate the query conditions.
- Its cosine ranking formula sorts the documents according to their degree of similarity to the query.

The vector space model (Salton et al., 1975) allows breaking the text files into vectors of words. Keywords or text descriptions can be substituted for graphics or multimedia. The content of static page views can be easily pre-processed by parsing the HTML and reformatting the information or running additional algorithms as desired.

The vector space model procedure can be divided into three stages:

- Document indexing where content bearing terms are extracted from the document text.
- Weighting of the indexed terms to enhance retrieval of documents relevant to the user.

- Ranking the document with respect to the query according to a similarity measure.

In the vector space model each document d is represented by a *term frequency* (TF) vector,

$$d_{tf} = (tf_1, tf_2, \dots, tf_m),$$

where tf_i is the frequency of the i th term in the document. Each term is given a weight based on its tf and inverse document frequency (IDF) in the document collection.

$$w_i = tf_i * \log\left(\frac{D}{df_i}\right)$$

i.e, Term weight,

df_i is the number of documents that contain the i th term

and $\log(D/df_i)$ term is known as the *inverse document frequency*, IDF_i .

IDF is used to normalise the documents of different lengths, so that each document vector is of unit length.

In the vector space model, the cosine similarity is the most commonly used method to compute the similarity between two documents d_i and d_j , which is defined to be

$$\cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|}$$

This measure becomes one if the documents are identical, and zero if there is nothing in common between them (i.e., the vectors are orthogonal to each other).

5. Development of the personalisation system at IIT Kharagpur

5.1 Aims and objectives

The IIT, Kharagpur, was set up in 1950 as the first in the chain of seven such institutes (recently the Govt. of India has established new similar Institutes at Hyderabad, Patna, Ahmedabad, Punjab and Rajasthan), is a premier centre for education and research in

engineering, technology and sciences in India. The Institute now has 31 academic departments/centres/schools of excellence with about 450 faculty, 2200 employees and 5000 students. The Institute offers undergraduate courses in 17 disciplines, postgraduate courses in 60 disciplines apart from doctoral programmes in most of the departments/centres/schools. It also offers higher education programmes in management, medical sciences and law with an emphasis on the application of technology in these fields (<http://www.iitkgp.ac.in>).

The Central Library of IIT, Kharagpur is in the forefront of adopting ICT for client-centred electronic information services in India, aided by a very rich and unique collection of e-journals, e-books, bibliographic/full-text electronic databases in stand-alone or networked CD/through web access, and so on (<http://www.library.iitkgp.ernet.in>). Many of these e-resources are available thanks to the Ministry of Human Resource Development's (MHRD's) Indian National Digital Library in Science and Technology (INDEST) consortium for leading engineering and technological institutions in India (Arora and Agrawal, 2003). The library collection currently stands at about 350,000 volumes including books, back-volumes of periodicals, microforms, videos, theses, patents and standards, with subscriptions to over 1000 journals and electronic access rights to over 7000 journals either through its own subscriptions or through the INDEST consortium.

The Personalised Selective Access Information (SAI) system is designed so that users can search the library database and the Web with more functionality, without expending their time and effort learning the intricacies of the various search interfaces developed by the different publishers and database providers. SAI also offers users the ability to create personal web links, interact with the library, use a protection mechanism to access their data, and have the web links checked and updated as necessary. SAI is designed for academic staff of IIT, Kharagpur to meet their personalised information requirements by optimising the range and depth of relevant information source retrieval to conduct various academic and research activities.

5.2 Methodology

SAI orders the search results so that the most relevant item is shown on the top of the page, and also it helps the user by providing the keywords of interest to refine the search. This is achieved by learning the user's interests by monitoring user activities while he/she is searching the Web. Unlike most web recommendation systems which rely on explicit users' feedback, SAI collects users' interests implicitly and search results are re-ordered according to their interests thus saving search time. SAI utilises proxy level collection, to overcome the disadvantages associated with server-level and client-level collections, to collect the web data for web usage mining. With due consideration to Boolean and probabilistic models as above, the vector space model is used in this study to convert data into a quantifiable form.

Initially contact details of academic staff in different departments of the Institute and their research interests are collected and stored in the database. Journals, e-books and other research material available on the Web are collected and stored in a database. When a user connects to the SAI web server through user login and password, the PHP (Hyper-text Preprocessor) embedded in HTML connects to the SAI database and retrieves the relevant personal information from the personal details table, areas of interests from the subject terms table, journal links from the journal table, books links from the books table and news links from the news table. Search engines such as Google, Altavista, the Electronic Library of the Institution of Engineering and Technology and the Institute of Electrical and Electronics Engineers (IEL), Association of Computing Machinery portal, Elsevier ScienceDirect etc., are placed in a drop-down box, so that the user can retrieve information using any database or search engine from one window.

5.3 Design and implementation of the SAI system

The SAI system uses client-server architecture and has been implemented using PHP and Java script. The database used, MySQL, is an open source SQL database and the operating system is Linux. SAI services can be accessed through a web browser such as

Internet Explorer and Netscape Navigator. The database consists of several modules such as:

- Login details: Username and Password information
- Personal Details: Name, Department, Phone and e-mail-id
- Journals module: Journal name and Journal website address
- Books module: Book name and http link to book site
- Subject terms module: Terms related to users' areas of interest
- News module and Personal links module.

Each of the journals, books, personal links, subject terms and news modules contain two tables. For instance, the master table of the journal module has information about the journal code, journal name and journal web link and the 'child' table represents the relation between the journal and the academic staff. When the user enters the information related to a journal, the existence of journal information in master table is checked and if it doesn't exist, the information is stored in the master and child tables. Books, personal links, subject terms and news modules also have a similar mechanism to append new resources to the system.

The SAI system under development has two phases:

- Automatic profile creation for the user
- Re-ordering of search results.

a. Automatic profile creation: this is an offline process, which runs continuously at the back end. User profile collection is an important constituent of SAI. The system has the capability to learn users' interest by monitoring their activities while they search the Web. The user profile is augmented using the data obtained from the proxy server and applying the web usage mining techniques. The system also presents the users with re-ordered search results, based on their interests. Users' interests can be collected in two ways,

- Explicitly asking the users to give their interests.

- Automatically collecting the users' interests by observing their browsing behaviour.

Since asking the user explicitly may not provide complete information, automatically obtaining the user's interest is the better process. A proxy server may be used to collect automatically the data of interest accessed by users. A proxy server stores all the users' access logs and each log entry in the proxy server has the following information as in Figure 1 and a sample log entry is provided in Figure 2.

Take in Figure 1

Figure 1. Details of information in proxy log

LogFormat "%h %l %u %t \"%r\" %>s %b \"% {Referer}i\" \"% {User-agent}i\"", where,

%h - the IP address of the client (remote host) which made the request to the server

%l - the identity of the client determined by identd on the clients machine

%u - the userid of the person requesting the document as determined by HTTP authentication.

%t - the time that the server finished processing the request [The format is 10/Oct/2000:13:55:36 -0700]

\"%r\" - the request line from the client is given in double quotes.

%>s - the status code that the server sends back to the client.

%b - the last entry indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be "-". To log "0" for no content, use %B instead.

\"% {Referer}i\" - This gives the site that the client reports having been referred from.

\"% {User-agent}i\" - The User-Agent HTTP request header which is the identifying information that the client browser reports about itself.

Take in Figure 2.

Figure 2. Sample Log entry

```
10.17.32.71 - - [13/Sep/2004:16:36:13 +051800] "GET
http://www.google.co.hu/images/logo_sm.gif HTTP/1.0" 200 4707
"http://www.google.co.hu/search?hl=hu&ie=ISO-8859-2&q=calcuttaweb.com&meta="
"Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt; FunWebProducts)"

10.17.32.205 - - [14/Sep/2004:11:07:56 +051800] "GET http://www.iitkgp.ernet.in/
HTTP/1.0" 200 15322 "-" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
```

The steps in this first phase of ‘automatic profile creation’ are:

- Web Page Creation: the initial phase involves identifying users' domains of interest and constructing a personal page for the user.
- Profile Creation: each user has to create their own login, providing personal details, for using the search result reorder system.
- Pre-process: parse the HTML page and PDF file, deleting the stop words or (non informative words) such as “a”, “an”, “this”, “for” etc, stemming the plural noun to its singular form and inflexed verb to its original form, deleting the html tags such as <HEAD>, <H1>, <BODY> etc.
- Term weight calculation: calculate term frequencies (TF), inverse document frequencies and term weights for each document using vector space model.
- Clustering: cosine similarity between each document is calculated and clustered by using k-means algorithm.
- User interest identification: the terms which are having their weights greater than some threshold value in the cluster which contains maximum documents are considered as users interests.

b. *Reordering of search results.* The steps in this phase are:

- search result page accessed by the user is interpreted;
- each document is opened and it is preprocessed;
- cosine similarity between each document and the user’s interest based on current browsing history is calculated;
- the most similar pages get ranked near the top of the list.

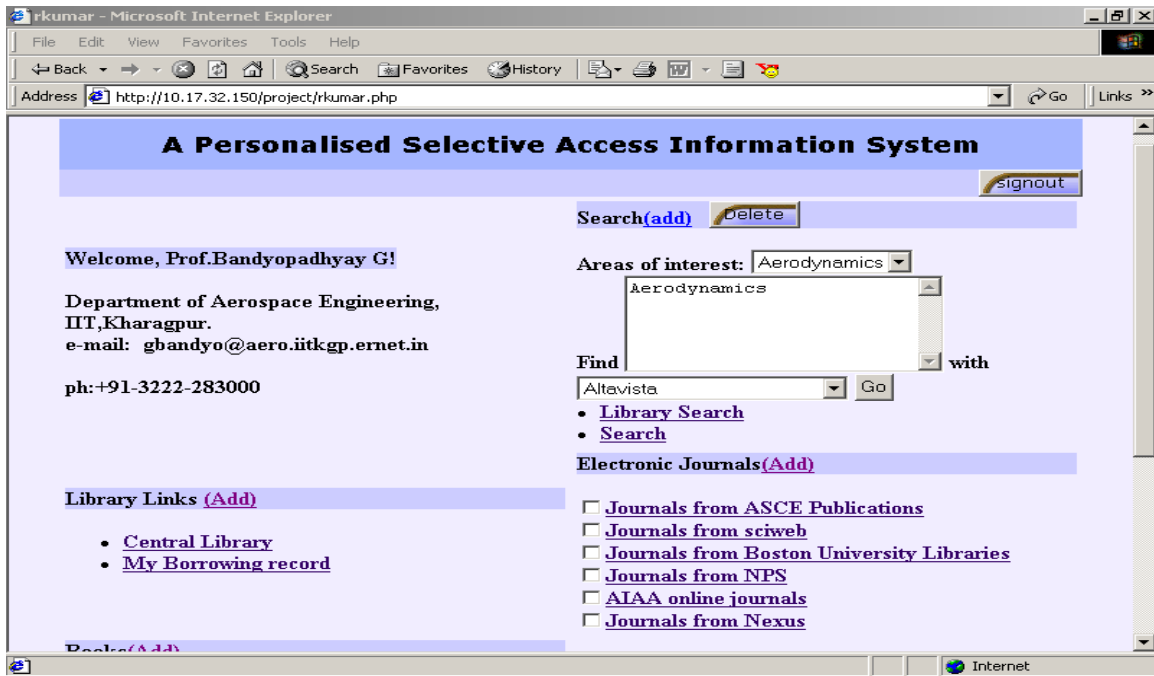
The features of SAI are as follows:

- Upon the initial login the user is presented with a personal page, whose contents and links are based on their subject specialisation.
- Welcome message with user's address, e-mail and phone number.
- Search: the user is provided with their areas of interest and multiple search engines. Just by a click on the area of interest the user can connect to the search engine of choice and get results displayed without explicitly typing the keyword.
- Library Links: a link to the library is provided to check information regarding books borrowed, corresponding due dates, overdue charged and any other person-centric queries from the library housekeeping database system.
- E-journals and e-books: the user is provided with links to e-journals and e-books related to their areas of interest. The user has a facility to add any additional links related to e-journals and e-books of choice in the 'add' page. The user is also provided with checkbox option to delete any of the existing links.
- News Links: the SAI design includes news links for users, which provides links about upcoming conferences worldwide, science and technology magazines, etc.
- Personal Links: facilitates addition of the links for the user's personal interests.
- Ask a Librarian: this provides a facility to use the suggestion form for the library or posting specific queries.
- Protect Mechanism: SAI has a protection mechanism based on username and encrypted password through which a user only can login to see the personalised page.

A sample page of SAI for one academic staff is presented in Figure 3.

Take in Figure 3.

Figure 3. Sample SAI User Page



5.4 Testing and results

The log data of three users from different departments collected during their search process are shown in Table I. the first column contains date and time of access of the URL, second column shows number of times the user has accessed the page, third column contains the IP address of the user machine by which the user can be recognised and fourth column provides the URL of the page visited. Data from this log data is taken and each URL is opened for term frequency and term weight calculations as shown in Table II.

Take in Table I

Table I Log Data

Date and time of access of the URL	Number of user access	IP address of the user machine	URL of the page visited
2005-06-15 11:58:44	1	10.17.32.2 05	http://www.cs.berkeley.edu/~balke/paper/enkompass02.pdf
2005-06-15 11:58:46	1	10.17.32.2 05	http://www.google.co.in/url?sa=T&ct=res&cd=12&url=http%3A//www.ercim.org/publication/ws-proceedings/DELOS5/straccia.pdf
2005-06-15 11:59:06	3	10.17.32.1 66	http://www.google.co.in/search?q=personalized+information+retrieval+systems&hl=en&lr=&ie=UTF-8&start=20&sa=N
2005-06-15 12:00:06	1	10.17.32.1 66	http://www.google.co.in/url?sa=T&ct=res&cd=21&url=http%3A//hypatia.slis.hawaii.edu/%7Elquiroga/Teaching/pidFA00/ics691announ.html
2005-06-15 12:00:16	1	10.17.32.1 67	http://www.google.co.in/url?sa=T&ct=res&cd=23&url=http%3A//lesk.com/mlesk/ages/ages.html
2005-06-15 12:00:23	1	10.17.32.1 67	http://www.google.co.in/url?sa=T&ct=res&cd=28&url=http%3A//www.Comp.rgu.ac.uk/staff/asga/intel.html

In Table II, the first column contains documents' id accessed by the user, second column shows the terms in that document, third column shows the term count, where term count is the number of times the term appears in the document, fourth column contains term frequency, fifth column contains the inverse document frequency and sixth column contains the term weight which is calculated by multiplying TF and IDF.

Take in Table II

Table II Term Weights

Doc. Id	Term	Term count	TF	IDF	Term weight = TF*IDF
D1	user	83	1.92427	5.78996	11.141500423981
D1	information	78	1.89762	5.78996	10.987185277787
D1	personalization	49	1.69897	5.78996	9.8369686566546
D1	users	42	1.63346	5.78996	9.4577172982229
D2	content	33	1.53147	5.78996	8.8672019322438
D1	services	26	1.43136	5.78996	8.2875391845462
D1	quot	26	1.43136	5.78996	8.2875391845462
D1	service	25	1.41497	5.78996	8.1926393276321
D1	used	20	1.32221	5.78996	7.6555970537011
D1	site	17	1.25527	5.78996	7.2679778081705

D1	uk	17	1.25527	5.78996	7.2679778081705
D1	needs	16	1.23044	5.78996	7.1242502471038
D1	provide	16	1.230448	5.78996	7.1242502471038
D1	profile	16	1.230448	5.78996	7.1242502471038

Table III shows term-document matrix with term weights which are used to form clusters. It is the matrix representation of terms and documents where row headings are terms and column headings are document ids. If a term appears in the particular document its weight is represented, otherwise it is represented by zero. Clusters are formed by calculating the similarities between each document by using cosine similarity or dot product.

Take in Table III

Table III: Term-document matrix with term weights

	Netw ork	optic	Wirel ess	Cap ac	Ada pt	Behav iour	Mod el	Fact or	Trav el	Attri bute	Quer y	Goo gle	Process or	energy
D1	3.33		2.408	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	6.22	3.33 9	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0.95 4	0	0	3.11 2	2.33 4	2.334	1.566
D4	0	0	0.602	3.89 0	3.11 2	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	15.2 67	0.95 4	42.7 9	31.9 0	0	0	0	0
D6	7	10.8 9	3.612	0	0	0	0	0	0	0	0	0	0	0

Table IV shows the clusters formed after using the clustering techniques. The first column in the table contains the cluster id and second column shows the number of documents in that cluster.

Take in Table IV

Table IV: Clusters

Clusters Id	No. of documents
0	1
1	2

2	3
---	---

Table V shows the users and users' interests obtained from the clusters. The table contains three users, where user 1 has only one document related to him and his interests obtained from this document are shown below. User 2 has three documents related to his area and user 3 has two documents related to his area. The first column in the table contains users, second column documents and third column shows the terms obtained from the clusters.

Take in Table V

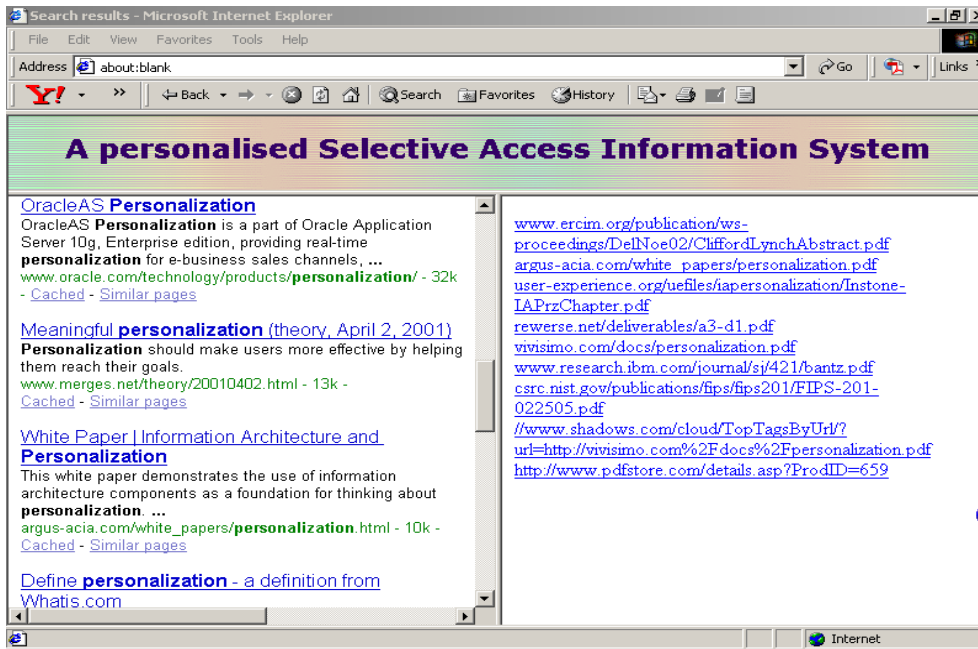
Table V: User interests from the clusters

Users	Documents	Terms obtained from the clusters
User1 (computer science)	1	query, google, processor, energy, factor
User2 (Electronics)	3	optic, network, wireless, capac, adapt
User3 (Civil)	2	behaviour, model, travel, attribute, value

The feedback received from the users show there is good agreement between the terms obtained by the system to represent the user interests. The reordered results in the search results show sufficient interest in the researchers. A categorisation of the relevance of terms and validity of results will be discussed in future. Figure 4 shows the sample screen of search results reordered for the query term 'Personalization'.

Take in Figure 4

Figure 4. Search results sample screen



6. Conclusions

There are practical constraints in the implementation stage. There is considerable difficulty in getting real and correct user interests and mapping them effectively into the products and services offered by the library. Also the interests of users keep on changing continuously. If multiple users share the same PC, it is difficult to identify the user as there is no one to one mapping between user and IP address. Continuously changing IP address also creates problems. Another potential roadblock is users' unwillingness to reveal personal information to fine-tune personalisation features due to privacy threats and misuse of personal data in the hands of doubtful elements.

The system developed has the capability to learn a user's interest by monitoring user activities while he/she searches the Web. It also has provisions to augment the user profiles using the data obtained from a proxy server and by applying the web usage mining techniques. The system also presents the user with reordered search results. The system is under various stages of testing both by the design team and end users to test its utility and wider applications in an academic information dissemination environment to predict usage and cross discipline specialisations of researchers. This application attempts

to add value to the results returned by normal search engines by re-ranking them in such a way that the user's context is reflected – so that different users still get the same set of results, but ordered in such a way that the results that are more relevant to each user are ranked higher. Note that these are not quite personalised search results but can be taken just as a personalised ranking of the standard results. To proceed further with more serious personalised results require effective mechanisms to trace and track what happens at the client side once they access the personalised results.

References (All URLs were checked on 6th October 2008)

Agrawal, R. and Srikant, R. (1994), “Fast algorithms for mining association rules in large data bases”, in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94), 12-15 September, Santiago, Chile*, pp. 478-499.

Anand, S. S. and Mobasher, B. (2005), “Intelligent techniques in Web personalization”, in Mobasher, B. and Anand, S. S. (eds.) *Intelligent Techniques in Web Personalization. Lecture Notes in Artificial Intelligence*, 3169, Springer-Verlag, Berlin, Germany, pp. 1-36.

Arora, Jagdish and Agrawal, Pawan (2003), “Indian Digital Library in Engineering Science and Technology (INDEST) Consortium: consortia-based subscription to electronic resources for technical education system in India: a Government of India initiative” in Murthy, T. A. V. et al. (Ed.), *Mapping Technology on Libraries and People, Proc. International CALIBER 2003, Ahmedabad, 13-15 February 2003*, INFLIBNET, Ahmedabad, pp. 271–290. Available at: <http://dlist.sir.arizona.edu/246/>

Belkin, N. J. and Croft, W. B. (1992), “Information filtering and information retrieval: two sides of the same coin?”, *Communications of the ACM*, Vol. 35 No. 12, pp. 29-38.

Catledge, Lara D. and Pitkow, James E. (1995), “Characterizing browsing strategies in the World-Wide Web”, *Computer Networks and ISDN Systems*, Vol. 27 No. 6, pp. 1065 - 1073.

Chen, Chien Chin, Chen, Meng Cheng and Sun, Yeali (2001), “A web Document Personalisation User Model and System”, in *Proceedings of User Modelling*. Available at <http://www.im.ntu.edu.tw/~paton/papers/conference/UM01WS-PVA.pdf>

Chen, L. and Sycara, K. (1998), “WebMate: A personal agent for browsing and searching”, in *Proceedings of the 2nd International Conference on Autonomous Agents, Minneapolis, Minnesota*, ACM Press, New York, pp. 132-139.

Cooley, R., Mobasher, B. and Srivastava, J. (1997), "Web Mining: Information and Pattern Discovery on the World Wide Web", in *Proceedings of the 9th International Conference on Tools with Artificial Intelligence (ICTAI'97)*, IEEE, USA, pp. 558-567.

Cooley, Robert, Mobasher, Bamshad, Srivastava, Jaideep (1999), "Data preparation for mining World Wide Web browsing patterns", *Knowledge and Information Systems*, Vol.1, pp. 5-32. Available at: <http://maya.cs.depaul.edu/~classes/ect584/papers/cms-kais.pdf>

Diebold, Boris and Kaufmann, Michael (2001), "Usage-based visualization of web localities", in *Proceedings of the 2001 Asia-Pacific symposium on Information Visualisation, Sydney*, Vol.9, pp.159-164.

Eirinaki, M. and Vazirgiannis, M. (2003), "Web mining for web personalization", *ACM Transactions on Internet Technology*, Vol. 3 No. 1, pp. 1-27.

Han, E., et al. (1998), "WebACE: A web agent for document categorization and exploration", in *Proceedings of the 2nd International Conference on Autonomous Agents, Minneapolis, Minnesota*, ACM, New York, pp. 408 – 415. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.8915>.

Han, J., Cai, Y. and Cercone, N. (1993), "Data driven discovery of quantitative rules in relational databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5 No.1, pp. 29-40.

Jeevan, V.K.J. and Padhi, P. (2006a), "A selective review of research in content personalization", *Library Review*, Vol. 55 No.9, pp. 556 - 586.

Jeevan, V.K.J. and Padhi, P. (2006b), "Preparedness for personalizing content in IIT libraries", *Electronic Library*, Vol. 24 No.5, pp. 680-693.

Joachims, T., Freitag, D., Mitchell, T. (1997), "WebWatcher: A tour guide for the World Wide Web", in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, pp. 770-776. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.9902>

Kramer, Joseph, Noronha, Sunil and Vergo, John (2000), "A user-centered design approach to personalization", *Communications of the ACM*, Vol. 43 No.8, pp. 44-48.

Lieberman, H. (1995), "Letizia: An agent that assists web browsing", in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, pp. 924-929. Available at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.3135>

- Manber, U., Patel, A. and Robison, J. (2000), "Experience with personalization of Yahoo!", *Communications of the ACM*, Vol. 43 No. 8, pp. 35 – 39.
- Robertson, S. E. and Sparck Jones, K. (1976), "Relevance weighting of search terms", *Journal of the American Society for Information Science*, Vol. 27 No. 3, pp. 129-146.
- Salton, G., Wong, A. and Yang, C. S. (1975), "A vector space model for automatic indexing", *Communications of the ACM*, Vol. 18 No. 11, pp. 613 – 620.
- Shahabi, Cyrus and Banaei-Kashani, Farnoush (2002), "A framework for efficient and anonymous web usage mining based on client-side tracking", in Kohavi, R., Masand, B., Spiliopoulou, M., Srivastava, J. (eds.), *WEBKDD 2001 - Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA*, Revised Papers, Lecture Notes in Computer Science, 2356, Springer, 2002, pp. 113-144.
- Srivastava, Jaideep, Cooley, Robert, Deshpande, Mukund and Tan, Pang-Ning (2000), "Web Usage Mining: discovery and applications of usage patterns from Web data", *ACM SIGKDD Explorations Newsletter*, Vol. 1 No. 2, pp. 12-23.
- Steinbach, Michael, Karypis, George and Kumar, Vipin (2000), "A Comparison of Document Clustering Techniques", available at <http://glaros.dtc.umn.edu/gkhome/fetch/papers/docclusterKDDTMW00.pdf> (accessed on 12 February 2008)
- Vozalis, E., Nicolaou, A., Margaritis, K.G. (2001), *Intelligent Techniques for Web Applications: Review and Educational Application*. Available at <http://macedonia.uom.gr/~mans/papiria/hercma2001.doc>
- Zhang, F. and Chang, H. (2002), "Research and development in web usage mining system—key issues and proposed solutions: a survey", in *Proceedings of the First IEEE International Conference on Machine Learning and Cybernetics Proceedings, Beijing, 4-5 November*, pp. 986-990.