2001

Clustering of Web Users Using Session-based Similarity Measures

Jitian Xiao

Edith Cowan University

Yanchun Zhang

University of Southern Queensland

10.1109/ICCNMC.2001.962600

This is an Author's Accepted Manuscript of: Xiao, J. , & Zhang, Y. (2001). Clustering of Web Users Using Session-based Similarity Measures. Proceedings of 2001 International Conference on Computer Networks and Mobile Computing . (pp. 223-228). Beijing, China. IEEE. Available here

Clustering of Web Users Using Session-based Similarity Measures

Jitian Xiao
School of computer & Information Science,
Edith Cowan University, Mount Lawley,
WA 6050, Australia,
j.xiao@ecu.edu.au

Yanchun Zhang
Dept. of Mathematics & Computing,
University of Southern Queensland,
Toowoomba, Qld 4350, Australia,
yan@usq.edu.au

Abstract

One important research topic in web usage mining is the clustering of web users based on their common properties. Informative knowledge obtained from web user clusters has been used for many applications, such as the prefetching of pages between web clients and proxies. This paper presents an approach for measuring similarity of interests among web users from their past access behaviors. The similarity measures are based on the user sessions extracted from the user's access logs. A multilevel scheme for clustering large number of web users is proposed, as an extension to the method proposed in our previous work [15]. Experiments have been conducted and the results have shown that our clustering method is capable of clustering web users with similar interests.

1. Introduction

The rapid web development and the increased number of available web searching tools push more and more organizations to put their information on the web and provide web-based services. In the meantime, the continuous growth in the size and use of the Internet is increasing the difficulties in searching for information. Reductions on the Internet traffic load and user access cost is therefore particular important.

One important research point in web usage mining is the clustering of web users based on their common properties. By analyzing the characteristics of the clusters, web designers may understand the users better and may provide more suitable, customized services to the users. One method to cluster web users is to measure similarity of interests between web users' access patterns and then cluster them based on the similarities obtained. By mining web users' historical access patterns, not only the information about how the web is being used, but also some demographics and behavioral characteristics of web users could be determined [4]. The navigation path of the web-users, if available to the server, carries valuable information about the user interests.

There has been an increased demand for understanding of web-users due to the Web development and the increased number of web-based applications. Based on deferent criteria, web users can be clustered and useful knowledge can be extracted from web user access pattern [12]. Many applications can then benefit from the knowledge obtained. For example, the *dynamic hypertext link generation* among web pages could be suggested after discovering clusters of web users that exhibit similar information needs [13]. This results in a better understanding of how users visit the web site, and leads to an improved organization of the hypertext documents for navigational convenience. Another application is the *prefetching* of web pages to help users to personalize their needs, reducing their waiting time [6]. Other applications to this kind of knowledge include *proxy cache* organization [2, 3] and mapping between user navigation paths [13].

There exist quite a few methods of clustering web users in literature [8, 9, 12]. However, the direct application of these methods on the primitive user access data is very inefficient and may not find interesting clusters because a web server may usually contains thousands even millions of pages, and web users may access web pages with a diversity of interests [9].

In our previous work [15], levels of similarities of web users are defined to capture different user's web-accessing interests. The definition of the similarity is application dependent. The similarity function could be based on visiting the same or similar pages, or the frequency of access to a page [1, 13], or even on the visiting orders of links (i.e., users' navigation paths). In latter case, two users

0-7695-1381-6/01 $10.00 © 2001 IEEE

that access the same pages could be mapped into different groups of interest similarities if they access pages in distinct visiting orders. A matrix-based algorithm is then developed to cluster web users such that the users in the same cluster are closely related with respect to the similarity measure. However, the performance of the clustering method becomes worse with the increase of the number of users when a threshold number is reached. Moreover, a user may visit the same web site many times using the same or different paths, which makes it hard to measure the visiting-order based similarity. To deal with the problems, we make two extensions to the previous work in this paper: Firstly, we propose to cluster web users using a multilevel clustering method, which is more suitable for clustering a large amount of web users. Secondly, the session-based similarity is adopted and the related clustering techniques are updated.

The rest of this paper is organized as follows: Section 2 defines the problem and some necessary concepts. In Section 3, we will review the matrix-based clustering method proposed in [15], and then present a multilevel scheme for large number of web users. Simulation results are presented in Section 4. And Section 5 concludes the paper.

2. Problem Definitions

For simplicity, we limit our concerns on the part of web users' navigation path inside a particular web site. From the Internet browsing logs, we could gather the following information about a web user: the frequency of a hyper-page usage, the lists of links she/he selected, the elapsed time between two links, and the order of pages accessed by individual web users.

2.1. User-Session identification

The task of identifying unique users is rather complicated by the existence of local caches, corporate firewalls, and proxy servers. Therefore some heuristics are commonly used to help identify unique users [8]. As the web access logs span a long period of time, it is likely that different users may use the same computer to access the websites. Thus, simply using the machine's IP address to identify unique users is quite problematic because multiple users may share one computer, e.g., many students may share a computer in an IT laboratory.

We differentiate the log entries into user-sessions, or simply *sessions*, through a session timeout. A *session* refers to the unit of interaction between a user and a web server [9]. It consists of pages accessed by a user in a certain amount of time. If the time between page requests exceeds a certain limit, it is assumed that there is another user-session, even though the IP address is the same. A web user's historical access pattern may have more than one session because he/she may visit a web site from time to time and spend arbitrary amount of time between consecutive visits.

Web user clusters are found based on sessions instead of the user's entire histories. The fact we cluster sessions instead of users can be justified that our goal is to understand the usage of the web and different sessions of a user may correspond to the visits with different purposes on mind. In addition, multiple users on a share computer can be represented by different sessions. In this paper, we will use session and user interchangeably.

The sessions are identified by grouping consecutive pages requested by the same user together. The data in a web server log is preprocessed to form a set of sessions in the form of (*session-id, {page-id, time}*), where session-id is a unique ID assigned to the session. A session (*s-id, p_0, 20, p_1, 30, p_2, 58*) tells a user spent 20 seconds on page p_0, 30 seconds on page p_1, and 58 seconds on page p_2.

The web server's log is scanned to identify sessions. A session is created when a new IP address is met in the log. Subsequent request from the same IP address is added to the session as long as the elapse of time between two consecutive requests does not exceed a pre-defined parameter *max-idle-time*, which is set as 30 minutes in our work. Otherwise, the current session is closed and a new session is created.

2.2. Session-based Similarity measures

Suppose that, for a given web site, there are m sessions $S = \{s_1, s_2, \ldots, s_m\}$ accessing n different web pages $P = \{p_1, p_2, \ldots, p_n\}$ in some time interval. For each page p_i and each session s_j, we associate a *usage value*, denoted as $use(p_i, s_j)$, and defined as

$$use(p_i, s_j) = \begin{cases} 1 & \text{If } p_i \text{ is accessed by } s_j \\ 0 & \textit{Otherwise} \end{cases}$$

The $use(*, *)$ vector can be obtained by retrieving the access logs of the site. If two users accessed the same pages in sessions, they might have some similar interests in the sense that they are interested in the same information (e.g., news, electrical products etc). The number of common pages they accessed can measure this similarity. The measure is defined by

$$Sim1(s_i, s_j) = \frac{\sum_k (use(p_k, s_i) * use(p_k, s_j))}{\sqrt{\sum_k use(p_k, s_i) * \sum_k use(p_k, s_j)}} \qquad (1)$$

where $\sum_k use(p_{k,} s_i)$ is the total number of pages that were accessed by the user of session s_i, and $\sum_k(use(p_{k,} s_i)* use(p_{k,} s_j))$ is the number of common pages accessed by both s_i and s_j. If two users access the exact same pages, their similarity will be 1. The similarity measure defined this way is called *usage based* (UB) measure.

Generally, the similarity between two users can be measured by counting the number of times they access the common pages at all sites. In this case, the measure is defined by

$$Sim2(s_i, s_j) = \frac{\sum_{k,w} (a_w(p_k, s_i) * a_w(p_k, s_j))}{\sqrt{\sum_{k,w} (a_w(p_k, s_i))^2 * \sum_{k,w} (a_w(p_k, s_j))^2}} \qquad (2)$$

where $a_w\ (p_k,\ s_i)$ is the total number of times that the user of session s_i accesses the page p_k at site w. (2) is called *frequency based* (FB) measure.

The similarity between two users can be measured more precisely by taking into account the actual time the users spent on viewing each web page. Let $t(p_k,\ s_j)$ be the time the user of session s_j spent on viewing page p_k (assume that $t(p_k,\ s_j\ ) = 0$ if s_j does not include page p_k). In this case, the similarity between users can be expressed by

$$Sim3(s_i, s_j) = \frac{\sum_k (t(p_k, s_i) * t(p_k, s_j))}{\sqrt{\sum_k (t(p_k, s_i))^2 * \sum_k (t(p_k, s_j))^2}} \qquad (3)$$

where $\sum_k\ (t(p_k,\ s_i))^2$ is the square sum of the time the user of session s_i spent on viewing pages at the site, and $\sum_k$ t$(p_{k,}\ s_i)$* t$(p_{k,}\ s_j)$ is the inner-product over time spent on viewing the common pages by users of s_i and s_j. Even if two users access exact same pages, their similarity might be less than 1 in this case, if they view a page in different amount of time. (3) is called *viewing-time based* (VTB) measure.

In some applications, the accessing order of pages by a user is more important than that of the time on viewing each page. In this case, two users (or sessions) are considered having the same interests only when they access a sequence of web pages in the exact same order. The similarity between users, in such a situation, can be measured by checking the access orders of web pages in their navigation paths. Let $Q = q_1,\ q_2,\ \ldots,q_r$ be a navigation path, where q_i, $1 \le i \le r$, stands for the page accessed in order. We call Q an r-hop path. Define Q_l as the set of all possible l-hop subpaths ($l \le r$) of Q, i.e., $Q_l = \{q_i,\ q_{i+1},\ \ldots,\ q_{i+l-1} \mid i = 1,\ 2,\ \ldots,\ r-l+1\}$. It is obviously that Q_l contains all pages in Q. We call $f(Q) = \cup_{l=1}^{r} Q_l$ the *feature space* of path Q. Note that a cyclic path may include some of its subpaths more than once, and $Q \subseteq f(Q)$.

Now let Q^i and Q^j are the navigation paths accessed by users in session s_i and s_j, respectively. The similarity between s_i and s_j can be defined using the natural angle between paths Q^i and Q^j (i.e., $\cos(\theta_{Q^i,Q^j})$), which is defined as:

$$Sim4(s_i, s_j) = \frac{<Q^i, Q^j>_l}{\sqrt{<Q^i, Q^i>_l \cdot <Q^j, Q^j>_l}} \qquad (4)$$

where $l = \min(length(Q^i),\ length(Q^j))$, and $<Q^i,\ Q^j>_l$ is the inner product over the feature spaces of paths Q^i and Q^j, which is defined as

$$<Q^i, Q^j>_l = \sum_{k=1}^{l} \sum_{q \in Q_k^i \cap Q_k^j} length(q) \cdot length(q) \qquad (5)$$

Based on the above definition, the similarity between two users (sessions) in terms of sessions will be 1 if they access a sequence of pages in the exact same access order, and 0 if they access no common pages at all. Note that $<Q^i,\ Q^j>_l$ is the same for all $l \ge \min(length(Q^i),\ length(Q^j))$. We call (4) the *visiting-order based* (VOB) measure.

The similarity between web users is application-dependent. If for some reason a more complicated similarity measure is needed, an applicable one could be defined for the individual application. For example, consider the similarity among the navigation paths between web users described by Shahabi et al. [12]. The navigation paths described in the paper are reproduced here as sessions (the names, in *italic*, in the paths are the title of web pages): (s_1, Main, *20*, Movies, *15*, News, *43*, Box-Office, *52*, News, *31*, Evita, *44*); (s_2, *Music*, 11, *Box-Office*, 12, *Crucible*, 13, *Books*, 19); (s_3, *Main*, 33, *Movies*, 21, *Box-office*, 44, *News*, 53, *Box-office*, 61, *Evita*, 31); (s_4, *Main*, 19, *Movies*, 21, *News*, 38, *Box-Office*, 61, *News*, 24, *Evita*, 31, *News*,19, *Evita*, 39); (s_5, *Movies*, 32, *Box-Office*, 17, *News*, 64, *Box-Office*, 19, *Evita*, 50); (s_6, *Main*, 17, *Box-Office*, 33, *News*, 41, *Box-Office*, 54, *Evita*, 56, *News*, 47).

The computation of similarity among web users' sessions results in an $m \times m$ matrix, called users' session-based *similarity matrix (SM)*. Assume that the above six sessions, identified by s_1, s_2,..., s_6, be the access traces of six users. By using the formula (1), the similarity between them is SM1 as shown below. The first and the third users visited the exact same pages, thus the similarity between them is 1 (i.e., SM1(1, 3) = 1). On the other hand, the similarity between the first and the second user is 0.224 (i.e., SM1(1, 2) = 0.224) because only one common page (i.e., *Box Office*) is accessed, although they visited five and four pages, respectively.

$$SM1=\begin{pmatrix} 1 & .224 & 1 & 1 & .894 & .894 \\ .224 & 1 & .224 & .224 & .25 & .25 \\ 1 & .224 & 1 & 1 & .894 & .894 \\ 1 & .224 & 1 & 1 & .894 & .894 \\ .894 & .25 & .894 & .894 & 1 & .75 \\ .894 & .25 & .894 & .894 & .75 & 1 \end{pmatrix}$$

Similarly, the similarity matrix based on (4) is as SM4.

$$SM4=\begin{pmatrix} 1 & .01 & .096 & .618 & .096 & .08 \\ .01 & 1 & .02 & .006 & .027 & .02 \\ .096 & .02 & 1 & .063 & .735 & .271 \\ .618 & .006 & .063 & 1 & .066 & .069 \\ .096 & .027 & .735 & .066 & 1 & .362 \\ .08 & .02 & .271 & .069 & .362 & 1 \end{pmatrix}$$

2.2. Data preprocessing

Sessions are extracted from web user access logs. The user access logs provide accurate, active and objective information about the web usages of the users. Moreover, most web servers contains such information in their log of page requests. Each record of the web server's log represents a page requests from a web user. A typical record contains the user's IP address, the data and time the request is received, the URL of the page requested, the protocol of the request, the return code of the server indicating the status of the request handling, and the size of the page if the request is successful. From such a web server log, user access pattern can be extracted, which consists of the pages the user visited and the time she/he spent on. Sessions can then be produced.

The data in the SM matrix also need to be preprocessed. In most cases, we are interested in those users (sessions) among them higher similar interests are shown with respect to a particular similarity measure. For this purpose, we could determine a *similarity threshold*, *t*, to split the user clusters. Users with similarity measure greater than or equal to *t* are considered in a same cluster. For instance, if t = 0.2, SM4 becomes the following SM4' after preprocessing.

$$SM4'=\begin{pmatrix} 1 & 0 & 0 & .618 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & .735 & .271 \\ .618 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & .735 & 0 & 1 & 0 \\ 0 & 0 & .271 & 0 & 0 & 1 \end{pmatrix}$$

The similarity measures can be useful for various web-based applications, even for improving the Web site performance. For example, based on the SM matrix, we are able to cluster web users into clusters such that the users in the same cluster are closely related with respect to the interest similarity measures. This clustering result can then be used for many applications [12].

3. A Matrix-based clustering algorithm

In this section, we will first review the matrix-based web user clustering algorithm proposed in [15], evaluate briefly the performance of it, and then propose a multilevel scheme for clustering large number of web users.

3.1. The matrix-based clustering algorithm

SM(i, j) represents the similarity measure between sessions (or users) s_i and s_j ($1 \leq i, j \leq m$). The greater the value of SM(i, j), the closer the two users s_i and s_j are related. Note that SM is a symmetric matrix and elements along the main diagonal are all the same (i.e., SM(i, i) = 1, $\forall\ 1 \leq i \leq m$). Thus only those elements in the upper triangular matrix need to be stored in implementation.

Clustering web-users into groups is equivalent to decomposing their SM matrix into sub-matrices. Our goal of clustering users (sessions) is achieved by two steps: (1) permute rows and columns of the matrix such that those "closely related"' elements are located closely in the matrix; and (2) find the dividing point that decomposes the matrix into sub-matrices. The clustering algorithm is detailed in [15]. The complexity of the algorithm is $O(m^2 log_2 m)$ where m is the number of sessions (or users).

3.2. Multilevel scheme for clustering large amount of web users

The matrix-based clustering algorithm works well for a small number of users (or sessions), say $m < 100$. With the increase of number of users (or sessions), the performance of it becomes worse. We now propose a multilevel scheme for clustering large number of web users.

The idea of multilevel clustering scheme is like this: For a large number of sessions, a session *similarity graph* (SG) is created whose node set consists of all sessions. If $SM(s_i, s_j) \neq 0$, an edge (s_i, s_j) is in the edge set with an edge weight of $w(s_i, s_j)= SM(s_i, s_j)$. The SG graph is first coarsened down to a *threshold* (say a hundred) number of nodes; a partition phase of this much smaller graph is applied; then the partition is projected back towards the original graph (finer graph); and finally each part of the finer graph is mapped to a similarity matrix, that can be clustered using the matrix-based clustering method. Formally, for a weighted graph $G_o=(V_0, E_0)$, with weights both on nodes and edges, the multilevel scheme consists of four phases.

Phase 1 (Coarsening): G_0 is transformed into a sequence of smaller graphs $G_1, G_2, ..., G_k$ such that $|V_0|>|V_1|>...>|V_k|$, with $|V_k|$ less than a pre-determined threshold t (say $t<100$).

Phase 2 (Coarse Partitioning): A partition P_k of the graph $G_k = (V_k, E_k)$ is computed that partitions V_k into q parts $V_{k1}, V_{k2}, ..., V_{kq}$, $q>1$, each containing about k/q nodes of G_0.

Phase 3 (Uncoarsening): The partition P_k of G_k is projected back to G_0 by going through intermediate partitions $P_{k-1}, P_{k-2}, ..., P_1, P_0$.

Phase 4 (Fine Partitioning): Each of $V_{01}, V_{02}, ..., V_{0q}$ is further partitioned (say by mapping to SM matrix and then use the matrix-based clustering method).

Phase 2 and Phase 4 can be implemented using the matrix-based algorithm discussed in previous sections, if the number of nodes of the input graph is no more than the predetermined threshold. Furthermore, if the number of the nodes of input graph in Phase 4 is still very large (say, greater than t), then the finer graph can be treated as an input graph of Phase 1 and a recursive procedure applies. We focus on the Phase 1 and the Phase 3 in the rest of this section.

During the coarsening phase, a sequences of a smaller graphs, each with fewer nodes, is constructed. Graph coarsening can be achieved in various ways [14].

At coarsening phase, a set of nodes of G_i is combined to form a single node of the next level coarser graph G_{i+1}. Let V_i^v be the set of nodes of G_i combined to form node v of G_{i+1}. We refer to node v as a multinode. The weight of node v is recomputed according to the weights of the nodes in V_i^v. Also, in order to preserve the connectivity information in the coarser graph, the edges of v are the union of the edges of the nodes in V_i^v. In the case that more than one node of V_i^v have edges to the same node u, the weight of the edge of v is equal to the sum of the weights of these edges. This is useful when we evaluate the quality of a partition at a coarser graph. The weight of the edge-cut[1] of the partition in a coarser graph will equal to that of the edge-cut of the same partition in the finer graph.

During the uncoarsening phase, the partition P_k of the coarser graph G_k is projected back to the original graph by going through the graphs G_{k-1}, G_{k-2}, ..., G_1, G_0. Since each node of G_{i+1} contains a distinct subset of nodes of G_i, obtaining P_i from P_{i+1} is done by simply assigning the set of nodes V_i^v collapsed to $v \in V_{i+1}$ to the partition $P_{i+1}[v]$ (*i.e.*, $P_i[u] = P_{i+1}[v], \forall u \in V_i^v$). We refer to this scheme as *non-refinement uncoarsening*. In order to get better projected partitions, refinement during uncoarsening phase is usually employed. Our refinement algorithm to uncoarsen coarser graphs to finer ones is based on the Kernighan-Lin's (KL) partition algorithm [11].

4. Simulations

The simulation is to demonstrate the capability of our clustering method for clustering Web users with similar interests. Using the obtained knowledge, we conducted another simulation to demonstrate the latency reduction of web document pre-fetching between caching proxies and browser users in our work (due to space limit, results of the second simulation is omitted).

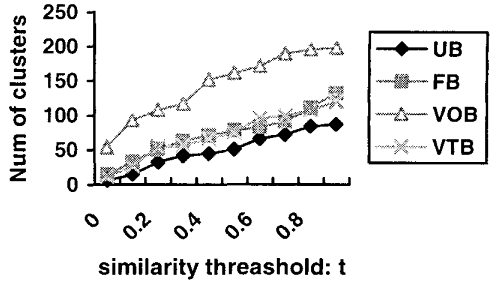

Fig 1. Clustering of 600 user-sessions.

In our simulation, we cluster web users using the multilevel scheme. The SG graph is produced using the session-based similarity measures computed from target data sets. Four similarity measures (i.e., UB, FB, VTB and VOB) are used and compared with each other. In the simulations, some data sets are generated, while others are extracted from the actual Internet access log files. For instance, we extracted some trace data from the BU-Web-Client trace, which can be freely

[1] An edge-cut of a graph is a set of edges whose removal will disconnect the graph

downloaded from Internet [5]. The traces contain records of the HTTP requests and user behavior of a set of Mosaic clients running in the Boston University Computer Science Department, spanning the timeframe of 21 Nov. 1994 through 8 May 1995. There are totally 1,143,839 requests of data transfer, from a population of 762 different users. Due to the memory limitation of our experimental setup, we extract some 600 sessions as our sampling data space.

During the simulations, we run 10 times at each simulation point. The number of y-axis is the mean values of all runs. Fig. 1 shows the results of clustering 600 user sessions. The similarity threshold t changes from 0.1 to 0.9. For a different t value, different number of clusters is produced. On average, the number of clusters produced for VOB-based measure is always greater than that of others, suggesting a finer granularity of clusters of users with similar interests. The number of clusters produced for UB-based measure is always the smallest among the four measures.

5. Conclusions

We presented a multilevel scheme for clustering web users using session-based similarities in order to capture the common interests among web users, which are characterized using different similarity measures. As a web user may visits a web site from time to time and spend arbitrary amount of time between consecutive visits, the web user clusters are found based on sessions instead of the user's entire histories. For some popular sites, the web servers may contains thousands even millions of pages, and web users may access web pages with a diversity of interests. Whenever the number of sessions is greater than a threshold, the sessions, and thus the related users, are clustered through a multilevel clustering scheme, otherwise they are clustered by the matrix-based clustering methods. Experiments have been conducted and the results have shown that our method is capable of clustering web users with similar interests.

References

[1] T. Bray, Measuring the Web. *Proc. of the Fifth International World Wide Web Conference*, Paris, France, May 1996.

[2] P. Cao and S. Irani, Cost-Aware WWW Proxy Caching Algorithms, *Proc. of the 1997 USENIX Symposium on Internet Technology and Systems*, Dec 1997.

[3] P. Cao, J. Zhang and K. Beach, Active Cache: Caching Dynamic Contents on the Web. *Proc. of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98)*, 1998

[4] L. D. Catledge and J. E. Pitkow. Characterizing Browsing strategies in the World Wide Web. *Electronic Proc. of the 3rd International WWW Conference*, Darmstadt, Germany, April 1995.

[5] C. A. Cunha, A. Bestavros and M. E. Crovella, Characteristics of WWW Client Traces, *Technical Report, TR-95-010*, Boston University Department of Computer Science, April 1995.

[6] C. R. Cunha, and C. F. B. Jaccound, Determining WWW User's Next Access and its Application to Prefetching. *Proc. of International Symposium on Computers and Communication'97*, Alexandria, Egypt, July 1997.

[7] L. Fan, P. Cao, W. Lin and Q. Jacobson, Web Prefetching between Low-Bandwidth Client and Proxies: Potential and Performance, *SIGMETRICS'99*, 1999.

[8] R. Cooley, B. Mobasher and J. Srivastava, Data Preparation for Mining World Wide Web Browsering Patterns, *Knowledge and Information Systems*. No.1 1999.

[9] Y. Fu, K. Sanghu and M-Y Shir, Clustering of Web Users Based on Access Patterns, Proc. of WEBKDD'99, San Diego, USA, 1999.

[10] S. D. Gribble, UC Berkeley Home IP HTTP Traces, July 1997, http://www.acm.org/sigcomm/ITA/ .

[11] G. Karypis and V. Kumar. Multilevel Graph Partition And Sparse Matrix Ordering. *Intl. Conf. on Parallel Processing*, 1995.

[12] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah, Knowledge Discovery from Users Web Page Navigation, *IEEE RIDE'97*, 1997.

[13] T. W. Yan, M. Jacobsen, H. G. Molina and U. Dayal, From User Access Patterns to Dynamic Hypertext Linking, *Proc. of the Fifth International World Wide Web Conference*, Paris, France, May 1996.

[14] J. Xiao, Y. Zhang & X. Jia. A Graph-based Multilevel Scheme for Reducing Disk Access Cost of Spatial Join Processing. *Proc. of the International Conference on High Performance Computing (HPC'2000)*, Beijing, China, May 2000, p823-830.

[15] J. Xiao, Y. Zhang, X. Jia & T. Li. Measuring Similarity of Interests for Clustering Web-Users. *Proc. of the* 12th *Australian Database Conference 2001 (ADC'2001)*. Gold Coast, Australia, 29 January – 2 February, 2001. pp107-114.