COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 4, No. 4, pp. 301-324, 2004

003

A DATA WAREHOUSING AND DATA MINING FRAMEWORK FOR WEB USAGE MANAGEMENT*

EDMOND H. WU†, MICHAEL K. NG‡, AND JOSHUA Z. HUANG§

Abstract. A new challenge in Web usage analysis is how to manage and discover informative patterns from various types of Web data stored in structured or unstructured databases for system monitoring and decision making. In this paper, a novel integrated data warehousing and data mining framework for Website management and patterns discovery is introduced to analyze Web user behavior. The merit of the framework is that it combines multidimensional Web databases to support online analytical processing for improving Web services. Based on the model, we propose some statistical indexes and practical solutions to intelligently discover interesting user access patterns for Website optimization, Web personalization and recommendation etc. We use the Web data from a sports Website as data sources to evaluate the effectiveness of the model. The results show that this integrated data warehousing and mining model is effective and efficient to apply into practical Web applications.

Key words: Data mining, Data warehousing, Web services, Website management

1. Introduction.

The rapid progress of our capabilities in data acquisition and storage technologies has led to the fast growing of tremendous amount of data generated and stored in databases, data warehouses, or other kinds of data repositories such as the World-Wide Web. On the other hand, many current and emerging data management applications require support for real-time analysis of large scale and continuously changing data streams, e.g., online monitoring user patterns in Websites. Hence, there is a great demand on designing innovative solutions for various data-intensive data mining and data warehousing applications.

Web usage mining is the application of data mining techniques to discover usage patterns from Web data, in order to understand and better serve the needs of Web-based applications. J. Srivastva et. al (2000) also propose a three-step Web usage mining process which are called preprocessing, pattern discovery, and pattern analysis. Many researchers have proposed different data mining algorithms for mining user access patterns or trends from the user access sessions. For instance, Mobasher et al. (1996) used association rules mined to realize effective Web personalization. Shen et al. (1999) suggested a three-step algorithm to mine the most interesting Web access

*(Eds.) Wai Lam, Rui-song Ye, Haiying Wang, and Jun Zhang. Research supported by HKRGC 7130/02P, 7046/03P, 7035/04P and 7035/05P and FRG/04-05/II-51.

†Department of Statistics & Actuarial Science, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: hcwu@hkusua.hku.hk

‡Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: mng@maths.hku.hk

§E-Business Technology Institute, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: jhuang@eti.hku.hk

associations. Zaiane et al (1998) proposed to apply OLAP and data mining techniques for mining access patterns based on a Web usage mining system.

Recently, Web applications such as personalization and recommendation have raised the concerns of people because they are crucial to improve customer services from business point of view, particularly for E-commerce Websites. Understanding customer preferences and requirements in time is a premise to optimize these Web services. The field of adaptive Websites is drawing attention from the community (perkowitz, 1999). One of the new trends in Web usage mining is to develop Web usage mining system that can effectively discover users access patterns and then intelligently optimize the Web services Recent studies (Berendt, 2002), (Nakagawa, 2003), (Wu, 2003) have suggested that the structural characteristics of Websites, such as the Website topology, have a great impact on the performance or efficiency of Websites. Hence, combining with the structure of Website, we can gain more interesting results for Web usage analysis. Understanding the user behavior is the first step to provide better Web services.

During the past two decades, database technologies have been developed very fast. Traditional databases store sets of relatively static records, such as Web-logs in Web servers. However, many current and emerging applications require the databases to support online analysis of rapidly changing data streams. Limitations of traditional database management systems in streaming data applications have raised the interests of many researchers. Different data mining algorithms for streaming data have been proposed with diverse infrastructure and domain applications. Recent research includes mining stream signatures and representative trends (Cortes, 2000), decision trees (Hulten, 2001), and regression analysis (Chen, 2002) etc. Therefore, the new generation Web usage mining system should also be designed to be capable of discovering changing patterns from a data stream environment with multi-type Web sources.

The research issue we focus in this paper is the problem of dynamic user patterns discovery from large-scale clickstreams in Websites. To solve this problem, our model focuses on how to handle multi-typeWeb data and monitor the changing patterns for analysis by using some novel mathematical models and statistical indexes. Based on the patterns discovered, we also propose practical solutions for Website optimization by reorganizing the Website content and its structure. In this paper, we present an efficient data model for aggregating user access sessions to effectively support different data mining applications. Based on the data model, we can easily perform various knowledge discovery tasks, such as association rule mining, sequential pattern mining, clustering, and Web usage predicting etc. Using these mining results, we can provide multiple solutions for various Web applications. For example, online personalized services, effective recommendation system, Website optimization etc.

This paper is organized as follows. In Section 2, we present the framework of a

multidimensional Web usage mining model. In Section 3, we introduce the implementation of the model, after that, experiment results are given. Then, we demonstrate practical Web applications in a real Website for optimizing Website services based on the model in Section 4. Finally, We give some conclusions and present our future work in Section 5.

2. Integrating Infrastructure for Web Usage Analysis. Since Web usage mining techniques have been widely used in various Web applications, it is necessary to develop an integrated platform for effective Web usage analysis. For this reason, we propose a multidimensional model to smoothly integrate Web data preprocessing, Website content and topology information. The framework can also easily combine different data mining algorithms to support different Web applications. In this section, we will introduce the main components of the model individually.

2.1. Data Preprocessing Module. Yang et. al (2003) introduced a data-cube model to contain the original access sessions for data mining from Web-logs. Based on it, we also investigated the practice of dealing with Web-log data streams (Wu, 2004). These work provided feasible preprocessing solutions to turn large volumes of Web logs into useful session information. So, the model designed can support both online and offline Web usage analysis.

2.1.1. Data Cleaning. The Web log datasets, which include the URLs requests, the IP addresses of users and timestamps, provide much of the potential information of user access behavior in a Website. Usually, we need to do some data processing, such as invalid data cleaning and user identification. Then, the original Web logs are transferred into user access session datasets for analysis. The Web log datasets (like server logs, cookies) contain useful information about the users' navigational behaviors. However, we need to do some preprocessing to turn the original Web log data into user access sessions. It will also affect the quality of Web usage mining.

Fig. 1 is a sample of Web-log records (the format of the sample Web-log is IIS 5.0, some system information is ignored). After preprocessing of these original Web-log data sets, we can use these user access sessions directly for further pattern discovery and data analysis in Websites.

```
GET /guangao/newsite/otherserver/espnchat.htm
2003-04-08 00:00:03 66.196.72.88 - 211.154.223.18 80
GET /Comment/Newscomment.asp NewsID=9632&TableName=News16
2003-04-08 00:00:08 202.108.250.198 - 221.154.223.18 80
GET /StaticNews/2000-07-25/News20a1638.htm
2003-04-08 00:00:08 61.153.18.234 - 211.154.223.18 80
GET /worldcup/worldcupnew.css
2003-04-08 00:00:09 210.22.5.36 - 211.154.223.18 80
GET /Imager/eye.swf
```

FIG. 1. *A sample of web-log data.*

2.1.2. Session Identification. We consider a Web log as a relation table T that is defined by a set of attributes $\mathcal{A} = \{A_1, A_2, ..., A_m\}$. Usual attributes include *Host, Ident, Authuser, Time, Request, Status, Bytes, Referrer, Agent* and *Cookie*. Assume that transactions generated by different users are identified by a subset of attributes $S \subset \mathcal{A}$. Let U be a set of user ids and $F : S \rightarrow U$ a function that maps each unique combination of values of S to an user id of U. Let $A_t \notin S$ be the Time attribute. We first perform the following two operations on T:

1. Use F to derive a new user ID attribute A_U in T.
2. Sort T on A_U and A_t.

T is transformed to T' after the two operations. Let $A_k(t_I)$ be the value of attribute A_k in the Ith transaction of T'. We then identify sessions according to the following definition:

DEFINITION 1. A session s is an ordered set of transactions in T' which satisfy $A_U(t_{I+1}) = A_U(t_I)$ and $A_t(t_{I+1}) - A_t(t_I) < \tau$ where $t_{I+1}, t_I \in s$ and τ is a given time threshold (usually 30 minutes).

2.2. Data Warehousing Module.

2.2.1. Cube Model for Historical Access Sessions. Conceptually, a session defined in Definition 1 is a set of ordered pages viewed in one visit by the same visitor. We define the number of viewed pages in a session as the length of the session. Each page identified by its URL is described by many attributes, including:

- Page ID;
- Page_Category: a classification of pages in a Web site based on the context of the page contents;
- Total_Time: the total time spent at a page;
- Time: the time spent at a page in a session;
- Overall_Frequency: the total number of hits at a page;
- Session_Frequency: the number of hits at a page in a session.

The values of these attributes can be computed from particular Web log files. A particular page in a session is characterized by its attribute values while the set of ordered particular pages characterizes a session.

Let P_{max} be the length of the longest session in a given Web log file. For any session with a length $P < P_{max}$, we define the pages of the session between $P+1$ and P_{max} as missing pages identified with the missing value "-". As such, we can consider that all sessions in a given Web log file have the same length.

In particular, if a session's length is significantly longer than most other sessions, we define the following provision to deal with such cases. The measure is that we try to separate the session into several sequential sessions. We set the length of each session no longer than P_{max}. If a session's length is P_{max}, it indicates that the sequential session is also one of the sequential sub-sessions of the long session. For example, a

session's length is 99, $P_{max} = 20$, then we separate the session into 4 sub-sessions with length 20 and 1 sub-session with length 19. Under the provision, special long sessions can be easily identified by the data model.

Given the above considerations, we define a data cube model for representing sessions as follows:

DEFINITION 2. A cube model is a four tuple $\langle S, C, A, \mathcal{V} \rangle$ where S, C, A are the sets of indices for three dimensions $(Session, Component, Attribute)$ in which

1. S indexes all identified sessions $s_1, s_2, ..., s_n$,
2. C consists of P_{max} ordered indices $c_1, c_2, ., c_{P_{max}}$ identifying the order of components for all sessions,
3. A indexes a set of attributes, $A_1, A_2, ..., A_m$, each describing a property of sessions' components, and
4. $\mathcal{V}$ is a bag of values of all attributes $A_1, A_2, ..., A_m$.

The order of session components is very important in the cube model while the orders of dimensions S and A are irrelevant. Each index $a_I \in A$ is associated with a pair $\langle AttributeName, DataType \rangle$. In this figure, we assume that sessions are sorted on the value of Length(s_I) where function Length(s_I) returns the real length of session s_I. The component dimension C denotes the entities (e.g., Web pages) in sessions, in other words, the identifiable Web resources in a Website.

DEFINITION 3. Let F be a mapping from $\langle S, C, A \rangle$ to $\mathcal{V}$ that performs the following basic operations on the cube model:

1. $F(s, c, a) = v$ where $s \in S, c \in C, a \in A$ and $v \in \mathcal{V}$,
2. $F(s_k, ., a_i) = V_{s_k, a_i}$ where V_{s_k, a_i} is session s_k represented by attribute a_i,
3. $F(., ., a_i) = V_{a_i}$ where V_{a_i} is a $p \times n$ matrix,
4. $F(., [c_i, c_{i+z}], a_i)$ returns a $z \times n$ matrix which represents a set of partial sessions.

The four basic operations can satisfy flexible queries based on users' needs. For instance, by using the fourth operation, we can search a set of sessions and the corresponding visitors who are in favor of the Web pages relating to English Premier League. For the same reason, we have Definition 4.

DEFINITION 4. Let "|" be a concatenation operator. $F(s_k, ., a_i) \mid F(s_{k+1}, ., a_i)$ attaches session s_{k+1} to session s_k.

With these basic operators defined on the cube model, data preparation for different analysis tasks can be greatly simplified. For example, we can use $F(., ., a_i)$ to take a slice for cluster analysis and use $F(s_k, ., a_i)$ to obtain a particular session described by a particular attribute for prediction.

Aggregation operations can also be defined on each dimension of the cube model. For example, sessions can be aggregated to clusters of different levels through clustering operations. Page values can be aggregated to categories using a classification scheme.

The Component dimension presents an important characteristic of the cube model. In this dimension, the visit order of the pages in a session is maintained. Because it uses component positions as variables instead of page ids as taken by others [9], it provides a regular and flexible matrix representation of page sequences which can be easily analyzed by existing data mining algorithms such as clustering and sequential association analysis.

The Attribute dimension allows the components of sessions to hold more information. For example, we can easily include time spent in each page in cluster analysis. From the these attributes, traditional Web log summary statistics such as the top pages by hits and spending time can be easily obtained.

2.2.2. Website Topology. Besides Web-logs, Website structure is another data source containing potentially useful information. Website topology is the structure of a Website. The nodes in a Website topology represent the Web pages with URL addresses and the edges among the nodes represent the hyperlinks among Web pages. Mathematically, a Website topology can be regarded as a directed graph. We assume that each pair of nodes are connect to each other by at least one path, that is, all Web pages in a Website can be visited from each other through at least one path.

Fig. 2 shows an example of a Website topology. All the Web pages are assigned with unique labels. A Website topology contains linkage information among the Web pages. The hyperlinks establish an informative connection between two Web information resources. The original design of a Website topology reflects the Website administrators' expectations of user access patterns. However, it may not be consistent with the actual expectations of visitors. Hence, a Website topology combining with visitors' access tracks can help us to understand the visitors' behavior.

Table 1 is the corresponding connection matrix of the Website topology in Fig. 2. For example, the value '1' of the entry AB represents the presence of a direct hyperlink from A to B; the value '0' of the entry AE represents the absence of a direct hyperlink from A to E. However, there is at least one path from A to E, such as ACHE. Such data matrices can be regarded as Website topology data sources, which are helpful for analyzing user patterns in Websites.

2.2.3. Topology Probability Model. In [13], we propose a probability model to measure the transition probabilities among the Web pages in a Website topology. Let us first consider the association probability between any two Web pages x_i and x_j in a Website. Suppose a given Website topology G contains n Web pages $X = \{x_1, x_2, \ldots, x_n\}$. We denote the number of outgoing hyperlinks from x_k by h_k for $k = 1, \ldots, n$. When an user finishes browsing the current Web page x_i, he or she may continue to visit one of the h_i Web pages connected to the current Web page or just exit the Website. Therefore, there are $h_i + 1$ choices for the user to select after visiting x_i. In our model, we assume that the probability of visiting x_j after having

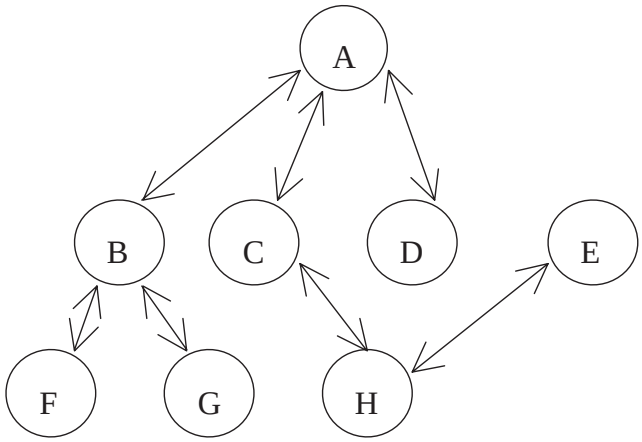

FIG. 2. *Website topology.*

TABLE 1
Connection matrix.

Page	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	0	0	0	1	1	0
C	1	0	0	0	0	0	0	1
D	1	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	1
F	0	1	0	0	0	0	0	0
G	0	1	0	0	0	0	0	0
H	0	0	1	0	1	0	0	0

visited x_i is given by:

$$P(x_j|x_i) = \begin{cases} \frac{w_{i,j}}{h_i+1} & \text{if there is a link from } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases}$$

Here, $w_{i,j}$ is a weighting parameter between x_i and x_j (usually we take $w_{i,j} = 1$). We also define the exit probability to be $P(\text{exit}|x_i) = 1 - \sum_j w_{i,j}/(h_i + 1)$. In particular, when $w_{i,j} = 1$ for all i, j, we have $P(x_j|x_i) = P(\text{exit}|x_i) = 1/(h_i + 1)$ where x_j and x_i are linked.

If a hyperlink from the page x_i to the page x_j exists, then we define the distance between x_i and x_j to be $D(x_i, x_j) = \log(1/P(x_j|x_i)) = \log((h_i + 1)/w_{ij})$. Otherwise, we consider a shortest path $x_1^* = x_i, x_2^*, \ldots, x_{m-1}^*, x_m^* = x_j$ from x_i to x_j. The distance between x_i and x_j is defined to be $D(x_i, x_j) = \log(1/P(x_1^* x_2^* \ldots x_m^*|x_i))$ where $P(x_1^* x_2^* \ldots x_m^*|x_i)$ is the probability of the shortest path given the starting point x_i.

Under the Markov assumption, $P(x_1^* x_2^* \dots x_m^* | x_i) = \prod_{k=1}^{m-1} P(x_{k+1}^* | x_k^*)$. Thus we may also express the distance measure as $D(x_i, x_j) = \sum_{k=1}^{m-1} D(x_k^*, x_{k+1}^*)$. We remark that the Website topology is a connected graph, thus, there must be a sequence of nodes connecting x_i and x_j. We can employ the classical Floyd algorithm to calculate the shortest paths between any two nodes in a Website topology.

Using the Website topology probability model and the distance measure, we can obtain information about the browsing patterns of users. On the other hand, we can also optimize the topology of a Website by minimizing a combination of the expected number of clicks and the number of hyperlinks in the Website [13].

2.2.4. The PUT-Cube Component. The PUT-Data Cube is the core of the multidimensional model. As we have mentioned, the Website structure information can greatly help us to analyze user patterns. Hence, we propose a data cube model which integrates Website topology, content, user and session information for multiple usage mining tasks. A novel cube model called PUT-Cube is defined as follows:

DEFINITION 5. A PUT-Cube model is a four tuple $\langle P, U, T, \mathcal{V} \rangle$ where P, U, T are the sets of indices for three main dimensions(Page, User, Time) in which

1. P indexes all page related attributes $P = P_1, P_2, ..., P_n$.
2. U represents of all user attributes $U = U_1, U_2, ..., U_m$ identifying users of groups or individual.
3. T indicates a set of temporal related attributes, $T = T_1, T_2, ..., T_r$, each describing the occurrence time or duration of user accesses.
4. $\mathcal{V}$ is a bag of values of all attribute combinations.

The PUT-Cube model focuses on most important factors in Web usage mining. If also considering the Website topology, the page factor not only provides information about which pages have been accessed, but also provides the information about their access order and relative positions. The user and time factors suggests the persons and time periods involve in the Web access events. Therefore, based on the 'who' ,'when' and 'which' user access information from the PUT-Cube, we can discover more useful user patterns, and then try to explain 'why'. This mining process is desired because we have concentrated on the primary factors for analyzing user behavior. The PUT-Cube model also provides the flexibility of selecting relevant dimensions(or attributes) for analysis.

The merit of the multidimensional model for Web usage management is that it combines multidimensional Web databases to support online or offline analytical processes. Based on the model, we can further propose and implement some solutions to intelligently discover interesting user access patterns for Website optimization, Web personalization and recommendation etc.

2.3. Data Mining Algorithm and Application Module. There are different kinds of data mining algorithms which can adapt in Web usage mining, such

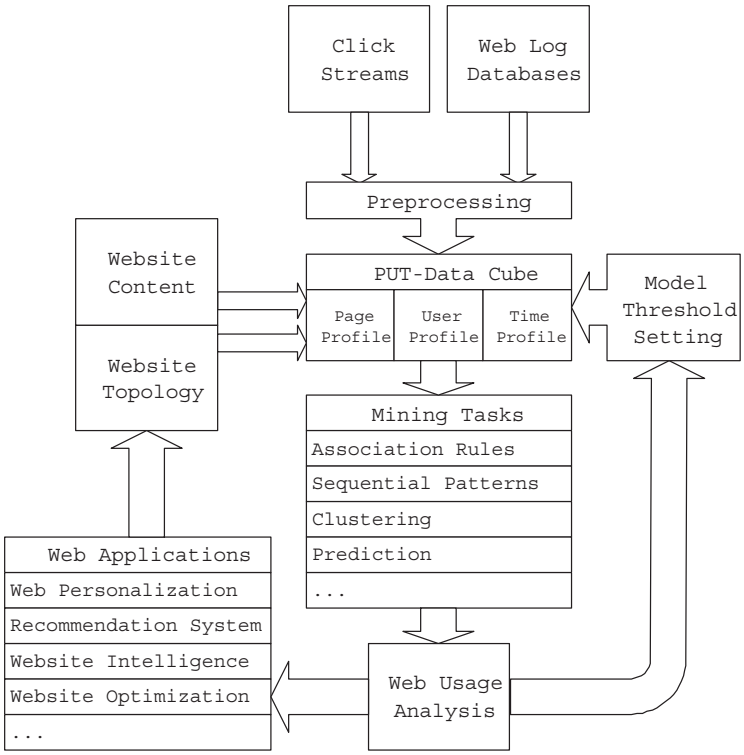

FIG. 3. *The multidimensional model for web usage mining.*

as association rule mining and clustering. For example, Wu et.al (2003) proposed a graph-based algorithm to find interesting association rules based on the Website topology. Our multidimensional model is capable of integrating these data mining algorithms efficiently. We can also set the model thresholds to see different mining results. Moreover, we can compare the results from different data mining algorithms for validation and deeper analysis. Then, we can improve the Website services by proposing different Web applications. As results, the Website content and topology should be changed to meet the needs of visitors. The effectiveness of the changes can also be validated under this framework. Fig. 3. shows the whole knowledge discovery process of the multidimensional model. We can see that the PUT-Cube module plays a key role in organizing the procedure of Web usage analysis.

3. Implementation and Experiments.

3.1. Model Implementation. In order to adapt the needs of different applications, we can set concept levels or specify value range to tailor a suitable model. For example, we can classify the Web pages into different categories, such as index page or content page. We may not need to include all the visitors in the 'U' factor. In fact, we usually select those users related to the user behavior analysis tasks, such as users

who visit frequently. Also, we prefer to set different temporal levels in the occurrence time T_I dimension, such as minute, hour, day and month, which also depend on the analysis tasks.

The following is an example of the model implementation. First, we select dimensions related to P-U-T factors. For example, in order to get the access matrix (like Table 3), we select P_1, P_2 where $P_1 = P_2 = V$, V is the set of Web pages in the Website topology. Each entry $C(i, j)$ represents the number of accesses from V_i to V_j. As to 'U', we adopt the set of User IDs for analysis. As to 'T', we set two different temporal dimensions. One suggests the time spent at the pages concerned. Another records an access occurrence time during given time intervals, such as minutes, hours etc.

In the above section, we introduce the CSA cube model (see Definition 2). We note that the main difference of PUT cube model and CSA model is that CSA cube model is mainly for Web data preprocessing, while PUT model focuses on the data analysis tasks. Therefore, the P-U-T factors and data are automatically selected from CSA model based on the analytical requirements.

Based on above analysis, we can implement a PUT-Cube model. The first is to aggregate the sessions into a multidimensional data cube model. Given an access session $S = \{S_1, S_2, ..., S_n\}$ by User k during time interval l, $T = \{T_1, T_2, ..., T_n\}$ index the time spent at each page in the session. The current Website topology is $G = \{V, E\}$, the corresponding topology probability matrix is P (refer to [13]). The session will be aggregated into the cube by the following operations:

(1) As to page access matrix, for each access S_iS_j, $C_k^l(S_i, S_j) = C_k^l(S_i, S_j) + 1$;
(2) As to page probability matrix, $P_k^l(S_i, S_j) = P(S_j|S_i)$;
(3) As to time duration matrix, $T_k^l(S_i, S_j) = T_i + T_j$;

The three equations above suggest how we implement the P-U-T factors in matrix forms. Based on these data obtained from the PUT model, we can detect user patterns and measure Website performance.

$C_{ij} = Count(S_i, S_j)$ indicates the number of accesses from S_i to S_j. We notice that the access session is the traversal path that an user follows in a Website topology. Any two adjacent pages in the session must be accessible by one click. Therefore, there are two cases of such counting, one is that S_i and S_j are adjacent in the access session S which also means the link of $S_iS_j \in E$. The other case is that S_i and S_j are not adjacent which means there exist at least one page between S_i and S_j. For the first case, we will count it whenever it appears. As to the second case, we will only count once in the access session. Based on the page counting scheme, Website analysts can capture the visiting sequential behavior as well as visiting preferences of Website visitors.

Example 1. Given access session $S = \{A, B, C, B\}$, in the first case, we will count AB, BC, and CB once since they are adjacent in the session, respectively. As

to the second case, we count AC once. But we won't count AB again since it has been counted once in the first case. So, $C(A,B) = 1$, $C(A,C) = 1$, $C(B,C) = 1$, $C(C,B) = 1$.

3.2. Cube Operations. Based on the cube model, we can explore our mining tasks to analyze the user behavior. we can select related dimensional attributes to form a cuboid lattice to discover interesting patterns. Cube operations, such as roll-up, drill-down, slice and dice can be easily performed in the data cube. Hence, the cube model is quite efficient to support different OLAP queries.

EXAMPLE 2. Table 2 is a sample user access session dataset which contains the access sessions of three users in six different time intervals in given site topology (See Fig. 2). Table 3 shows the aggregating matrix which sums up the accesses of three users during the specified time intervals. From the matrix, we observed that the most frequent access is AC which has total 9 accesses.

TABLE 2
User Access sessions.

	User1	User2	User3
Time1	EHC	ABFBG	ABGB
Time2	ABF	ACHEC	ACH
Time3	ACHE	GBA	ACHEHC
Time4	AD	CH	ABF
Time5	DACH	ABFBG	ABGB
Time6	EH	ADACHEHC	ACHC

TABLE 3
Access matrix.

Page	A	B	C	D	E	F	G	H
A	0	7	9	2	5	4	4	8
B	3	0	3	0	0	4	5	3
C	4	4	0	0	5	4	2	8
D	3	1	3	0	3	1	1	3
E	4	5	4	0	0	4	2	4
F	0	2	0	0	0	0	2	0
G	3	3	3	0	0	0	0	3
H	4	4	6	0	5	4	1	0

3.3. Model Performance Analysis. We use the Web usage data from ESPN-STAR.com, a sports Website, to test and evaluate the performance and effectiveness of our PUT-Cube Model proposed. With permission, we also get the topology of the Website for analysis.

We use two months Web log data to do the experiments. The original Web logs contain millions of access records from the Web servers. After data preprocessing, we got the user access session datasets. In some datasets, we take all the sessions during a period of time (e.g., one day). We also select the sessions from particular users for analysis in some datasets. Table 3 shows some of the datasets for experiments. ES1, ES2 and ES3 are the access session datasets from the logs during December, 2002 and ES4 and ES5 are the logs from April, 2003.

TABLE 4
Real web datasets.

Dataset	No.Accesses	No.Sessions	No.Visitors	No.Pages
ES1	583,386	54,300	2,000	790
ES2	2,534,282	198,230	42,473	1,320
ES3	6,260,840	517,360	50,374	1,450
ES4	78,236	5,000	120	236
ES5	7,691,105	669,110	51,158	1,609

EXPERIMENT 1. We first evaluate the efficiency of aggregating access sessions by PUT-Cube. Here, we use ES3 to test the running time when increasing the number access sessions. From the results shown in Fig. 4, we can see that the increase of running time exhibits a linear relationship with the increase of access sessions.

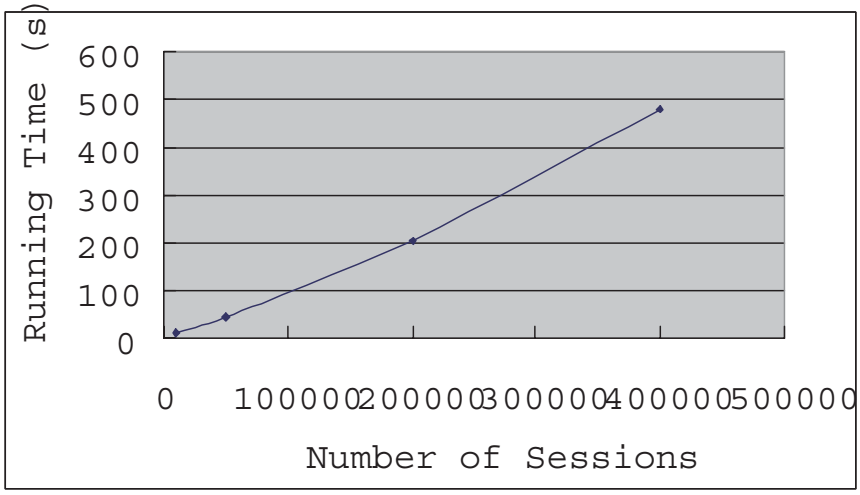

FIG. 4. *Increasing number of access sessions.*

EXPERIMENT 2. The second experiment is to evaluate the efficiency of the PUT-Cube when increasing the cardinality. We use datasets ES5 for testing. The result shows in Fig. 5. Through above analysis, we noticed that even for a Website with thousands of different pages, users or time intervals for analysis, the PUT-Cube model

is still feasible. We can get the results in an acceptable period of time.

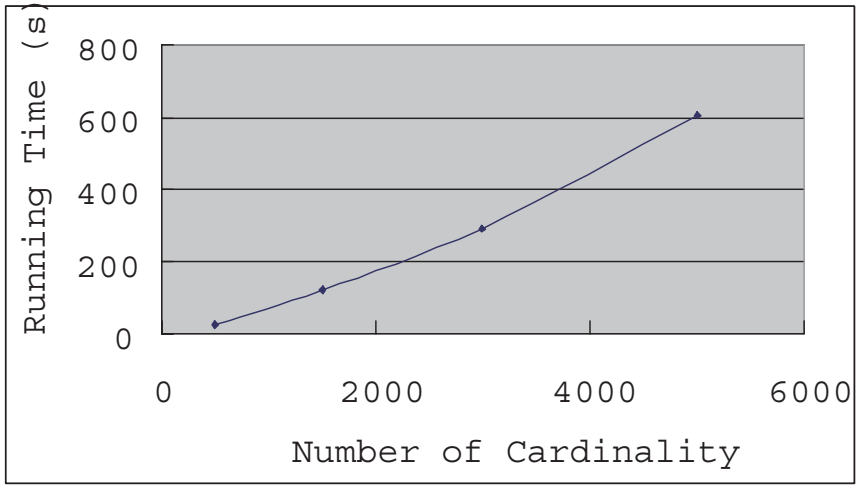

FIG. 5. *Increasing cardinality.*

EXPERIMENT 3. The third experiment is to test the efficiency of the PUT Cube model when increasing the number of dimensions. We choose the largest dataset ES5. The result shows in Fig. 6. The result shows that the cube model still work well even we include most of the session attributes in the cube model. In practice, we usually choose two to five dimensions for analysis. Hence, the cube model can work well for most analysis tasks, even for online analysis. We note that we define a three dimensions model in Definition 5. Based on the definition, we can further propose other dimensions for analytical applications. The new dimensions can be derived from the PUT model. For instance, the system performance dimension.

4. Web Applications. In this section, we demonstrate several practical application based on the multidimensional model combining different data mining algorithms proposed.

4.1. Website Optimization. We first propose our solution for Website optimization. In this paper, Website optimization refers to reorganize the Website topology and content to improve the access efficiency and system performance.

4.1.1. Measure of Discovery of Access Patterns. In [14], we suggest a novel measure named Access Interest (AI) as below:

DEFINITION 6. Given Aggregating Session Matrix C and Topology Probability Matrix P, page staying time T, the Access Interest Matrix is given by:

$$AI(i,j) = Log(\frac{C_{ij}T_{ij}}{P_{ij}} + 1) \tag{1}$$

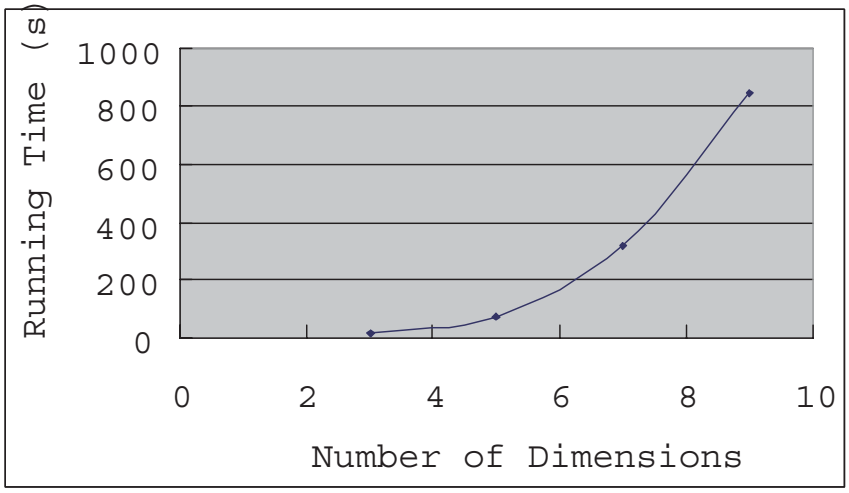

FIG. 6. *Increasing number of dimensions.*

where C_{ij} is the number of accessing from Web page V_i to V_j, and P_{ij} is the probability from V_i to V_j, T_{ij} is the average stay time of visiting V_i and V_j. Given $T = T_1, T_2, ..., T_n$, T_i is the average staying time of V_i. We define $T_{ij} = T_i + T_j$.

EXAMPLE 3. Using the sample dataset in Table 1, we discovered top 5 AI values in the given site topology (See Fig. 2): $AI(E, F) = 13.1, AI(H, F) = 12.5, AI(G, H) = 11.7, AI(E, G) = 11.4, AI(C, F) = 11.2$, respectively. From the results, we can conclude that these access patterns are important, however, the access efficiency is not satisfying because these sequential accesses are time consuming. For example, many visitors prefer to visit page F after visiting page E, but these two pages are not 'close' to each other in the Website topology. Therefore, discovering such patterns will be of great help to improve the Website access efficiency no matter from the user or system point of views. Hence, the results validated the effectiveness of using the cube model to mine exceptional user access patterns.

4.1.2. Indexes of Access Efficiency in Websites. From above analysis, we need to find new measurements to evaluate the Website access situation under a Website topology. For a given Website topology, visitors must follow certain traversal paths to access the Web pages that they are interested in. For instance, if a visitor wants to sequentially visit Web page {A, F, E} (See Fig. 2), the shortest traversal path is {A, B, F, B, A, C, H, E}. The corresponding access sequence is $S = \{AB, BF, FB, BA, AC, CH, HE\}$. Thus, the visitor should click at least seven times to access the target pages {A, F, E}. An access P1 P2 is defined as the access from page P_1 to P_2. If a visitor wants to browse the same target pages in a different order {A, E, F}, another traversal path {A, C, H, E, H, C, A, B, F} with eight clicks is needed, the corresponding accesses are {AC, CH,HE, EH, HC, CA, AB, BF}. But

if one want to access other 3 pages {A, B, G}, just two accesses are enough.

We observe that the access efficiency of {A, F, E} is low due to the redundancy of accesses. In general, there are two types of access redundancy. One is the jump-track access. For example, starting from A, the target page is F, we must access B first before we access F. The other type of redundancy access is the backtrack access, e.g., in order to access C from B, a visitor must click the back button in the browser to go from B to A and then to C, even though A has been accessed previously.

The difference between the jump-track access and backtrack access is determined by looking into whether the access has been performed in a visitor access session, e.g., in accessing {A, F, E}, AB is a jump-track access while BA is backtrack access. In this paper, we call both of them non-target accesses, comparing with target accesses which acquire the target Web pages directly. Because jump-track accesses are recorded in Web logs while back-tracks accesses won't be recorded by Web logs, therefore, a non-target access can only be either jump-track access or back-track access.

In practice, we will use the Web usage data (e.g., Web logs, cookies) as well as Website topology to distinguish between jump-track accesses and back-track accesses. For jump-track accesses, we need to separate target and non-target Web pages based on staying time of Web pages. For example, if a visitor only browses a Web page for 5 seconds, the page is not likely to be a target page. Usually, user accesses to index pages are classified as jump-track accesses.

As to back-track accesses, we need to use the Website topology to acquire access path information. For example, if a visitor enters homepage A (see Fig. 2), and wants to visit B and C sequentially. Then, she must revisit A after visiting B before she can visit C under the Website topology. However, Web logs will not record the access from B to A, therefore, we need to use the Website topology to recover this back-track access. Here, we assume that visitors won't use the URL address to access Web pages direct, they use the links in each Web page or back-button in browsers to access other Web pages. If visitors use the URL address to access Web pages, we regard such accesses as independent sessions.

In order to measure the access efficiency of a visitor access session for a Website topology, we define the User Access Efficiency (UAE) as follows:

DEFINITION 7. Given an user access session $S = \{s_1, s_2, , s_m\}$, $A = \{a_1, a_2, , a_i\}$ is the jump-track access session, $B = \{b_1, b_2, , b_j\}$ is the back-track access session, where $A \subset S$ and $B \subset S$, $A \cap B = \Phi$. The User Access Efficiency (UAE) is given as follows:

$$UAE(S) = 1 - \frac{|A| + |B|}{|S|} \tag{2}$$

where $|S|$ is the length of session S.

If Web pages are cached in the Web browser, the Web server will not record the backtrack accesses in the Web-log. For example, a visitor follows a traversal path {A, C, H, E, H, C, A, B, F}, the Web-log will only contain {A, C, H, E, B, F}, which is not a complete traversal path. Therefore, we propose a new measure called Server Access Efficiency to evaluate the access efficiency from thee Web server side of view.

DEFINITION 8. Given an user access session $S = \{s_1, s_2, , s_m\}$, $A = \{a_1, a_2, , a_i\}$ is the jump-track access session, $B = \{b_1, b_2, , b_j\}$ is the back-track access session, where $A \subset S$ and $B \subset S$, $A \cap B = \Phi$. The Serve Access Efficiency (SAE) is given as follows:

$$SAE(S) = 1 - \frac{|A|}{|S| - |B|} \tag{3}$$

where $|S|$ is the length of session S.

EXAMPLE 4. Given target Web pages T={A, E, F}, eight accesses S={AC, CH, HE, EH, HC, CA,AB, BF} are needed. AC, CH, AB are jump-track accesses and EH, HC, CA are backtrack accesses. The remaining two accesses HE, BF are target accesses. Hence, UAE(S)=1-(3+3)/8=25%,SAE(S)=1-3/(8-3)=40%.We can also calculate the access efficiency of a group of access sessions. Given an access session database D, the UAE and SAE of D are calculated as follows:

$$UAE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} UAE(S_i) \tag{4}$$

$$SAE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} SAE(S_i). \tag{5}$$

We notice that $SAE(S) \geq UAE(S)$, it can be explained that UAE will consider the effect of backtrack clicks to the access efficiency, while SAE will not count. We note that both UAE and SAE measures have practical meanings. If the UAE is low, it means that an user is hard or slow to access Web pages they wanted. If the SAE is low, it suggests that the Web server return too many irrelevant or uninteresting Web pages to the end users. If the access session S belongs to a frequently requested pattern, then it will cause the inconvenience of a large portion of visitors. What's more, if this happens at the peak hours of a day, much more requests will be sent to the Web server, the overload of Web server will significantly slow down the users accesses to the Website. The situation will result in the loss of visitors. Hence, it is particularly important to improve the User Access Efficiency and Server Access Efficiency according to the concerns of improving the satisfaction of visitors or maintaining the robustness of Web servers.

We remark that if a Website topology is a complete graph, then we always have the highest access efficiency. However, such topology is implausible for large websites. In our design, we would like to construct a new topology that can achieve better access efficiency than the original one without increasing the number of links among Web pages. The new topology is more adaptive to the user access patterns.

In the calculation of UAE and SAE, we assume that there are some target pages in each session. The assumption is compatible with the prevailing hierarchical structure of a Web site. Usually, we can identify most of the target pages based on the visiting time and categories of Web pages. With more information about user visiting habit or content information, we can get higher accuracy of identifying target pages.

4.1.3. Algorithm for Website Optimization. The Website topology optimization procedure contains four major steps. We summarize them as follows: The first step is to mine the interesting access pattern model described above. In the second step, using the interestingness measures AI to identify interesting access patterns for analysis. In third step, optimizing the Website topology based on the interesting access patterns discovered. In the last step, we can choose UAE and SAE to validate the effectiveness of the optimization results(refer to [14]).

Since we have found some interesting access patterns which represent the low access efficiency, we can build some direct hyperlinks to connect them. For instance, if we found E to F is a low access efficiency pattern and there is no direct hyperlink between them, then we can build a new link in page E to F. As results, the access efficiency from E to F will be greatly improved. Here is the description of the algorithm:

1. Input Web access sessions database D and Website topology $G = (V, E)$
2. Mining interesting access patterns and gain AI matrix using PUT-Cube model
3. Input top k access patterns $P_1Q_1, , P_kQ_k$with highest AI values and top m access patterns $P_1R_1, , P_kR_m$with lowest page access $C(P_j, R_j)$
4. For each pattern P_iQ_i, $i = 1, ..., k$, if $P_iQ_i \notin E$, build hyperlink P_iQ_i to connect the sequential access pattern. For each pattern P_jR_j, $j = 1, ..., m$, if $P_jR_j \in E$ and $AI(P_j, R_j)$ is low , disconnect hyperlink P_jR_j.
5. If the access efficiency of user patterns is satisfying, output the optimized Website topology

Example 5. Fig. 7 shows the optimized topology returned by the topology optimization algorithm with linkage optimization. Comparing with the original topology (See the left of Fig. 7), there are several significant changes in the structure of the topology. Recall the top 5 interesting access patterns EF, HF, GH, EG and CF, which can not be efficiently accessed in the original topology, however, in the new topology, they are quite efficient for the users to access.

In the optimized topology (See the right of Fig. 7), B and C have the common linkage, if we merge B and C into a new page, the number of the links required in the optimal topology is approximately equivalent to the original one.

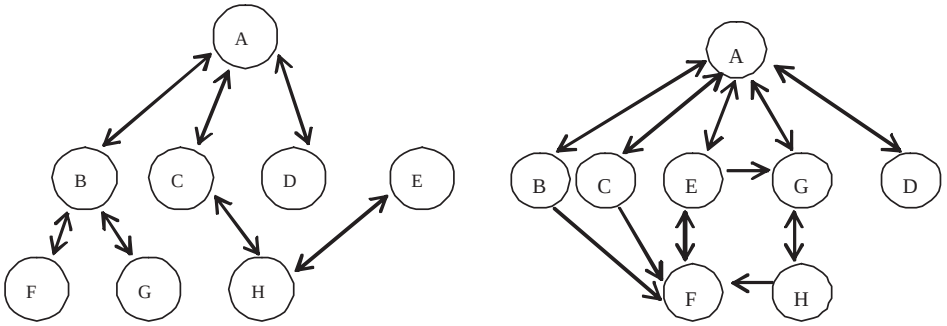

FIG. 7. *Comparison of original and optimized website topology.*

We should note that the optimization algorithm is feasible to apply in both global optimization and local optimization. That is, we can optimize the whole Website or part of the topology concerned.

4.1.4. A Real Case Study. We investigated the effectiveness of Website optimization model through a real case study. The server logs and Website topology is from ESPNSTAR.com.cn, a well-known E-commerce sports Website in China. The professional sports website has a focused set of interesting topics related to sports, such as football, basketball, and even golf. Hence, it has a very broad audience. Fig. 8 and Fig. 9 show the number of visitors and pages requested from a sports Website ESPNSTAR.com at each hour on April 1, 2003. We can see that the number of visitors and pages requested on each hour is not even. In peak hours, the pages requested can be five times of off-peak hours. Whats more, we noticed that during the ongoing period of some important matches, such as World Cup, thousands of visitors will focus on browsing certain pages interested in around the same time. It causes the traffic problems of the Web servers. Hence, some real time connectivity optimization solution is needed to improve the Website access efficiency during the peak hours.

An interesting discovery is that many visitors tend to visit the Website during some important matches, such as English Premier League, to look for related content. However, their access patterns change very quickly. As results, the problem occurred was that large volumes of page requested made the Web servers unstable, and hence lower the access speed. Through our analysis about Website optimization, we note that unnecessary pages requested can be decreased if the Website topology can well adapt to the current user access patterns. By the using Website topology algorithm based on the cube model, it is supposed that we can improve the access efficiency of Website.

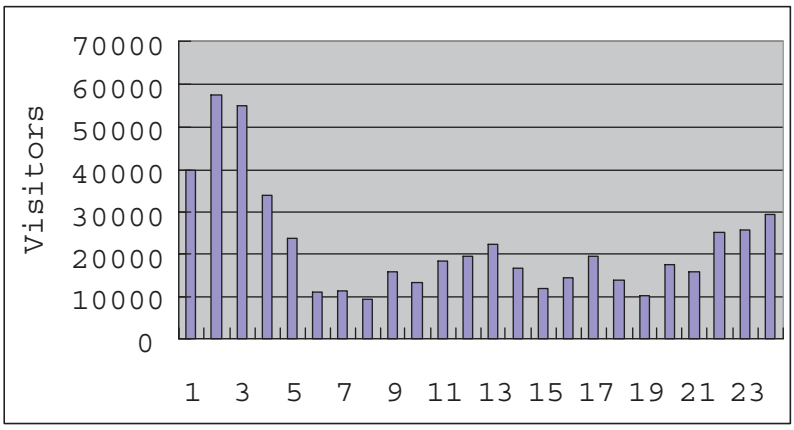

FIG. 8. *Distribution of visitors.*

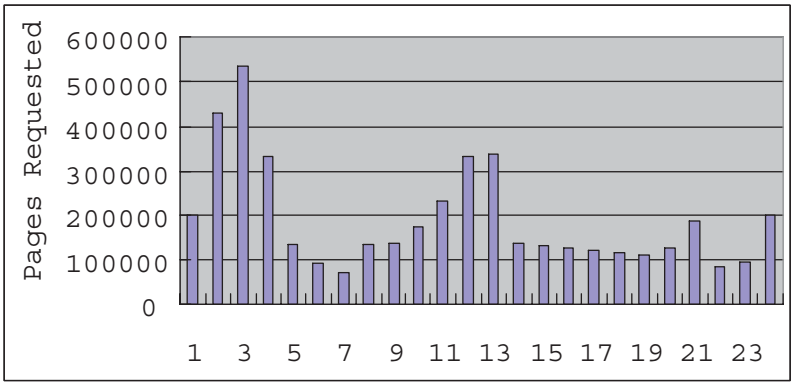

FIG. 9. *Distribution of pages requested.*

Fig. 10 is the comparison of access efficiency measures by original and optimized Website topology. We employed the ES5 dataset to do the experiment. The time period is from 0:00 AM to 4:00 AM, April 1, 2003. We can see that both the access efficiency measures of the optimized Website topology have great improvement than the original one, especially in the peak hours around 2 AM to 3 AM. It can be explained that during the peak hours, more people follow similar access patterns, for example, searching for hot news, on-going matches etc. The optimization model can help the Website to achieve more stable access efficiency. On average, it can have over 50% of improvement in access efficiency. That is, people may save a lot of time to search the content that they are interested in from the Website. The experiment validated the effectiveness of Website optimization based on our multidimensional model for Web usage mining. Hence, the model proposed is feasible to put into practice for Website optimization.

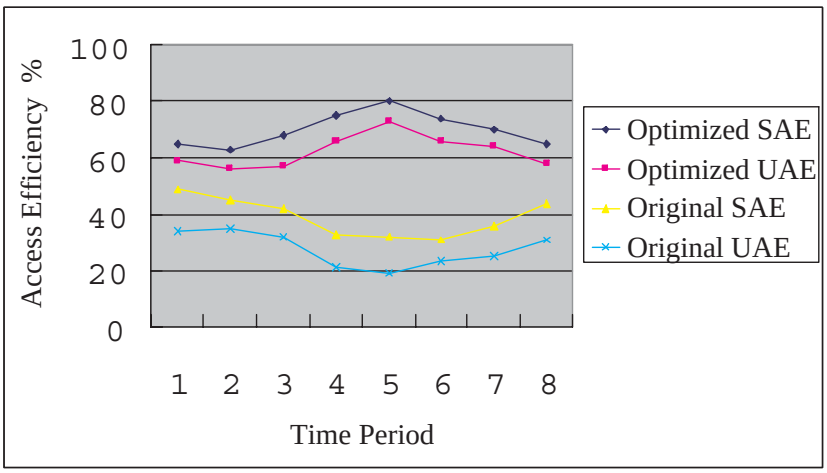

FIG. 10. *Connectivity optimization.*

4.2. Website Personalization. Other Web applications, such as personalization can also employ the multidimensional model to optimize the services. For example, we can use the cube model to find the access habits of a particular group of users or individual. With the help of domain experts, we proposed different cuboids to investigate user behavior. Several analytical cuboids are listed as follows:

- Cuboid_1:{Page_Category,Time_Occurrence, User_ID}
- Cuboid_2:{Page_Frequency,Time_Interval, User_Category}
- Cuboid_3:{Page_Frequency,Page_Topology, Time_Duration,User_Category}
- Cuboid_4:{Page_Frequency,Time_Occurrence, Time_Duration,User_ID}

These cuboids can help us to analyze the Website access situations from different point of views. In our investigation to a sports Website, we found that as to Website analysts, they were quite eager to know which content pages are frequently visited during which period of time, particular user prefers to what kind of sports content. The cube model can achieve these analysis tasks easily. As to Cuboid_1, we can analyze a particular user's visiting preference varying page contents and time occurrence. Cuboid_2 provides the basic user access statistics of the Website. Employing Cuboid_3, we can monitor the user patterns and access efficiency of Website. Using Cuboid_4, we can employ other data mining techniques, such as cluster analysis and temporal association rules.

Based on the mining results, the personalization system based on the model can intelligently build a personalized Website based on the user's interesting access patterns. For example, the personalization system can intelligently post the headlines which the user will probably be interested in. For football fans, the personalized system can automatically select the most exciting football news for them in the headlines (See the pointer 1 of Fig. 11).

FIG. 11. *Connectivity optimization.*

4.3. Recommendation System. The model can also be employed in recommendation system. For example, we found an interesting access pattern that many visitors in the Website would like to browse the scoreboards and calendar of European football leagues very frequently, e.g., English Premier League. However, the related contents scatter in the original Website. Discovering such pattern, the Website designers put related content in the homepage, so users can easily to acquire these information (See the pointer 2 of Fig. 11). Based on the individual access patterns, we can also promote some new sports activities or sports products to the users who may be interested in (See the pointer 3 of Fig. 11).

4.4. Web User Behavior Analysis. Web users behavior during their visiting to Website can provide valuable insights into the effectiveness of the Website, as well as Web users browsing patterns or preferences. The core value of identifying Web user behavior is to improve the quality of interaction of users and Website. A Web users action can be motivated by various needs, and these needs may change from moment to moment during a browser session. Due to this nature, effective behavior analysis is a critical and daily activity for Web masters. This is the most important Web application of our data warehousing and data mining framework. Our multidimensional model proposed can provide useful and realtime information of

users behavior from many aspects (e.g. design and implement task-driven analytical cuboids under the model proposed).

Some suggestions have been adopted in the Website redesign (Fig. 11 suggests the redesign of homepage which provides Web personalization and recommendation services). The dataset ES 4 contains the access activities of 120 members to their fee-paid services. By analyzing the customer behavior based on the cube model, the Website managers and designers can reorganize the content and its structure to provide better services. The cube model can also be employed in marketing research, CRM analysis etc. As results, these Web services can greatly improve customers' satisfaction.

Furthermore, we can propose some intelligent solutions for realtime user patterns discovery. For instance, we are interested in detecting the user patterns to the sports Website during different time periods of a sporting event, e.g., before, during and after a NBA basketball match. Based on the user patterns discovered from the Web usage data mining and warehousing framework, we can reorganize the Website's content and services for easy accesses by visitors. As a practical application, the intelligent system can online post a match's TV time-schedule in an attractive homepage position before a match and update the scoreboard quickly after a match.

To solve the system performance problem, we can adopt the PUT model to discover unusual patterns and take some measures to improve the access efficiency of users. What's more, as a Website security application, Website administrators can use the proposed indexes and patterns discovery methods to detect hackers' attacks to the Website when overflow of unusual access patterns has occurred. Therefore, the data mining and warehousing model can also be integrated as a daily system maintenance tool.

5. Conclusion. In this paper, we propose an efficient multidimensional data warehousing and data mining model for combining user access sessions with Website topology for Web usage analysis. The model focuses on the page, user and time attributes to form a multidimensional cube which can be frequently updated and queried. The real value of the Web usage management framework is that is can be used to effectively identify Web user behavior and then improve the quality of interaction of users in the Website. The experiments show that the data model is effective and flexible for different analysis tasks. Our real case studies suggest that the Web usage mining model can be applied in different Web applications, such as Website optimization, Web personalization, recommendation systems, and Web user behavior analysis. In the future, we intend to implement the multidimensional framework with more data mining algorithms and Web applications in an analytical data Warehousing system.

REFERENCES

[1] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou, *The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis*, Proceeding of the WEBKDD 2002 Workshop, Edmonton, Canada, 2002.

[2] Y. Chen, G. Dong, J. Han, B. W.Wah, and J. Wang, *Multi-Dimensional Regression Analysis of Time- Series Data Streams*, in: Proc. Int. Conf. on Very Large Data Bases, 2002, pp. 323–334.

[3] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava, *Websift: The web site information filter system*, in: Proceedings of the Web Usage Analysis and User Profiling Workshop, 1999.

[4] C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith. Hancock, *A Language for Extracting Signatures from Data Streams*, in: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining, 2000, pp. 9–17.

[5] J. Dougherty, R. Kohavi, and M. Sahami, *Supervised and Unsupervised Discretization of Continuous Features*, Proceedings of International Conference on Machine Learning, Tahoe City, CA, 1995, pp. 194–202.

[6] G. Hulten, L. Spencer,and P. Domingos, *Mining Time- Changing Data Streams*, in: Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining, 2001, pp. 97–106.

[7] B. Mobasher, N. Jain, E. Han, and J. Srivastava, *Web mining: pattern discovery from World Wide Web Transactions*, Tech. Rep. 96-050, University of Minnesota, Sep. 1996.

[8] Miki Nakagawa and Bamshad Mobasher, *A Hybrid Web Personalization Model Based on Site Connectivity*, WEBKDD, 2003.

[9] O. Nasraoui, H. Frigui, A. Joshi, and R. Krishnapuram, *Mining Web access logs using relational competitive fuzzy clustering*, Proceedings of the Eight International Fuzzy Systems Association Congress, 1999.

[10] M. Perkowitz and O. Etzioni, *Adaptive Websites: Conceptual cluster mining*, in: Proc. 16th Joint Int. Conf. on Artificial Intelligence (IJCAI99), pages 264–269, Stockholm, Sweden, 1999.

[11] L. Shen, L. Cheng, J.Ford, F.Makedon, V. Megalooi-konomou, and T. Steinberg, *Mining the most interesting web access associations*, Proc. of the 5th International Conference on Knowledge Discovery and Data Mining (KDD99), 1999, pp. 145–154.

[12] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan, *Web Usage Mining: Discovery and applications of usage patterns from web data*, SIGKDD Explorations, 1(2000), pp. 12–23.

[13] Edmond H. Wu, Michael, and K. Ng, *A Graph-based Optimization Algorithm for Website Topology Using Interesting Association Rules*, Proc. the Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2003), Seoul, Korea, 2003.

[14] Edmond H. Wu, Michael, K. Ng, and Joshua Z. Huang, *On improving website connectivity by using web-log data streams*, Proc. of the 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004), Jeju, Korea, 2004.

[15] Edmond H. Wu, Michael K. Ng, and Joshua Z. Huang, *An efficient multidimensional data model for Web usage mining*, Proceeding of the Sixth Asia Pacific Web Conference (APWEB2004), 2004.

[16] Q. Yang, J. Huang, and M. Ng, *A data cube model for prediction-based Web prefetching*, Journal of Intelligent Information Systems, 20(2003), pp. 11–30.

[17] Andy M. Yip, Edmond H. Wu, Michael K. Ng, and Tony F. Chan, *An efficient algorithm for dense regions discovery from large-scale data stream*, Proc. of the 8th Pacific-Asia

Conference on Knowledge Discovery and Data Mining (PAKDD2004), 2004.

[18] Osmar R. Zaiane, Man Xin, and Jiawei Han, *Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs*, in: Proc. ADL'98 (Advances in Digital Libraries), Santa Barbara, April 1998.