A Cube Model for Web Access Sessions and Cluster Analysis

Zhexue Huang, Joe Ng, David W. Cheung
E-Business Technology Institute
The University of Hong Kong
jhuang,kkng,dcheung@eti.hku.hk

Michael K. Ng*, Wai-Ki Ching
Department of Mathematics
The University of Hong Kong
mng@maths.hku.hk

Abstract

Identification of the navigational patterns of casual visitors is an important step in online recommendation to convert casual visitors to customers in e-commerce. Clustering and sequential analysis are two primary techniques for mining navigational patterns from Web and application server logs. The characteristics of the log data and mining tasks require new data representation methods and analysis algorithms to be tested in the e-commerce environment. In this paper we present a cube model to represent Web access sessions for data mining. The cube model organizes session data into three dimensions. The COMPONENT dimension represents a session as a set of ordered components $\{c_1, c_2, ..., c_P\}$*, in which each component* c_i *indexes the* i*th visited page in the session. Each component is associated with a set of attributes describing the page indexed by it, such as page ID, page category and view time spent at a page. The attributes associated with each component are defined in the ATTRIBUTE dimension. The SESSION dimension indexes individual sessions. In the model, irregular sessions are converted to a regular data structure to which existing data mining algorithms can applied while the order of the page sequences is maintained. A rich set of page attributes is embedded in the model for different analysis purposes. We also present some experimental results of using the* k*-modes algorithm to cluster sessions. Because the sessions are essentially sequences of categories, the* k*-modes algorithm designed for clustering categorical data is proved effective and efficient. Furthermore, we present a new approach to using the first-order Markov transition frequency (or probability) matrix to analyze clustering results for categorical sequences. Some initial results are given.*

Index Terms– Web usage mining, Web log analysis, Web data model, Clustering

*Research supported in part by Research Grant Council Grant Nos. HKU 7147/99P and 7132/00P, and HKU CRCG Grant No. 10203501.

1 Introduction

Web server log is a primary data source in Web mining [3][12]. A Web server log records transactions of connections to the Web server [13]. Each transaction presents an interaction between the Web server and a client machine a user used to visit the Website. Standard data elements in a Web log file include *Host, Ident, Authuser, Time, Request, Status* and *Bytes* [20]. Additional elements such as *Referrer, Agent* and *Cookie* can also be found in some log files. Details of these Web log data elements are given in [11]. At some popular Web sites, such as www.amazom.com and www.yahoo.com, millions of transactions are being generated daily in their log files. Managing and mining the huge Web data source has become a big challenge to many online organizations.

In Web mining, objects of different abstraction levels, such as *Users, Server Access Sessions, Episodes, Clickstreams and Page Views* [18] [11], are usually identified from Web log files for different mining tasks. Preprocessing techniques for extracting these objects are discussed in [2]. Among others, server access sessions (or sessions for short) are very important for *Web structure mining, Web usage mining* [12], and *path traversal patterns mining* [1]. Web usage mining is related to the problems of discovery of Web access patterns, online customer behavior analysis, Web personalization, and design and building of adaptive Web sites [10]. Clustering and sequential analysis of Web access sessions play a key role in solving these problems.

A Web server session is defined by W3C as a collection of user clicks to a single Web server during a user session or visit [21]. A simple view of a session is a sequence of ordered pages $\{P_1, P_2, P_3, P_4, P_3, P_5, \cdots,\}$ where P_i is a unique id for a page URL in the Web site. A session database can be viewed as a set of such sequences. Because different numbers of pages often occur in different sessions, this simple representation of sessions does not satisfy the input data requirement of many existing data mining algorithms. Adding more page information such as the time spent and total hits further complicates the session representation. Different approaches have been adopted to represent sessions for different mining tasks. In [14], a set of sessions is encoded as a set of N_U-dimensional binary attribute vectors in which each attribute represents a URL and its value is "1" if the URL appears in a session and "0" otherwise. This representation satisfies many existing data mining algorithms. The major concern is the lost of page orders. By using the attribute-oriented induction method [5], sessions with different numbers of pages can be generalized into vectors in the same dimensions based on the page hierarchy [4]. Regular vectors can be obtained after sessions are aggregated up to a certain level so clustering algorithms such as BIRCH [23] can be applied to the generalized sessions. One problem is again the lost of page orders. Another problem is the lost of information on the low level pages in the page hierarchy which are probably more interesting to users. The WUM system uses the sophisticated aggregate tree model to represent sessions which supports user initiative queries to discover navigation patterns using an SQL like query language MINT [17]. Such a structure can speed up the query process. However, data-driven mining process is not supported.

In this paper, we introduce a cube model for representing sessions to effectively support different mining tasks. The cube model organizes session data into three dimensions. The *Component* dimension represents a session as a set of ordered components $\{c_1, c_2, .., c_p\}$, in which each component c_i indexes the ith visited page in the session. Each component is associated with a set of attributes describing the page indexed by it. The attributes associated with each component are defined in the *Attribute* dimension of the cube model. Depending on the analysis requirements, different attributes can be defined in the Attribute dimension such as Page ID, Page Category and View Time spent at a page. The *Session* dimension indexes individual sessions. The details of the cube model are given in the next section. In comparison with other representation methods mentioned before, the cube model has the following advantages: (1) it represents sessions in a regular data structure to which many existing data mining algorithms can be applied; (2) it maintains the order of the page sequences and (3) it can easily include more page attributes for different analysis purposes. Simple operations can be defined to extract necessary data from the model for different mining operations and produce reports for summary statistics and frequency counts of page visits. Therefore, the model can be used as an efficient and flexible base for Web mining.

We present a result of using the k-modes algorithm [6] to cluster sessions described as sequences of page URL IDs. Since the URL IDs should be treated as categorical values, clustering such categorical sessions is clearly a challenge to come numeric only algorithms such as CLARANS [15] and BIRCH [23] which are incapable as pointed out by other researchers [9]. Our result has shown that the k-modes algorithm is efficient and effective in discovering interesting clusters with strong path patterns. We will present cluster analysis on two Web log files and some interesting clusters identified using the criteria of average distance of sessions to cluster centers and the size of clusters. We propose a Markov transition frequency (or probability) matrix approach to validating clusters of sessions and show some initial results.

This paper is organized as follows. In Section 2, we describe the cube model and some basic operations. In Section 3, we discuss the basis of the k-modes algorithm. In Section 4, we present a new approach to using the first-order Markov transition frequency (or probability) matrix to analyze clustering results for categorical sequences. In Section 5, we show some initial clustering results of two Web log files using the two clustering algorithms. We draw some conclusions and present our future work in Section 6.

2 A Cube Model to Represent User Sessions

2.1 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 a 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).*

Host IP address or Domain Name is often used in S to identify users [16] [17] but host IP address alone can result in ambiguity in user identification caused by firewalls and proxy servers. More attributes such as Referrer and Agent can be used to resolve this problem [2].

2.2 The Cube Model

Conceptually, a session defined 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.

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

Definition 2 *A cube model is a four tuple $< S, C, A, \mathcal{V} >$ 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 p_{max} ordered indices $c_1, c_2, .., c_{p_{max}}$ identifying the order of components for all sessions, and*
3. *A indexes a set of attributes, $A_1, A_2, ..., A_m$, each describing a property of sessions' components.*

$\mathcal{V}$ is a bag of values of all attributes $A_1, A_2, ..., A_m$.

Figure 1 (left) illustrates the cube model. 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 $< AttributeName, DataType >$. 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.

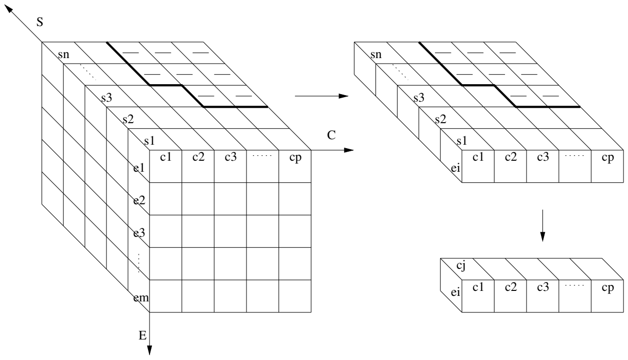


Figure 1. The cube model (left) and some operations (right).

Definition 3 *Let F be a mapping from $< S, C, A >$ 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 a_i attribute,*
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.*

Definition 4 *Let "$|$" be a cancatenation operator. $F(s_k, ., a_i) \;|\; F(s_{k+1}, ., a_i)$ 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 (Figure 1 (right)) and use $F(s_k, ., a_i)$ to obtain a particular session described by a particular attribute for prediction (Figure 1 (right)).

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 [14], 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 as we will show in the following sections. From the these attributes, traditional Web log summary statistics such as the top pages by hits and spending time can be easily obtained.

3 The k-modes Algorithm for Clustering Categorical Sessions

If we slice from the cube model only the Page variable in the Attribute dimension, we obtain a matrix which contains sessions described in categorical values. We can use the k-modes algorithm to cluster these categorical sessions. In this section we brief the k-modes algorithm. The experimental results of using k-modes to cluster categorical sessions extracted from two real Web log files will be given in Section 5. Furthermore, if we can also slice more variables such as Page ID and Time and form a session matrix of mixture data types. In this case, we can employ the k-prototypes algorithm [6] that is designed for mixture data types. However, in this paper, we focus on categorical sessions while the results of clustering sessions in mixture data types will be discussed elsewhere.

The k-modes algorithm is a variant of the k-means algorithm for clustering categorical data. It has made the following modifications to the k-means algorithm: (i) using a simple matching dissimilarity measure for categorical objects, (ii) replacing the means of clusters with the modes, and (iii) using a frequency based method to find the modes to minimize the following objective function

$$J_c(W,Z) = \sum_{l=1}^{k}\sum_{i=1}^{n} w_{li} d_c(Z_l, X_i) \tag{1}$$

subject to

$$\sum_{l=1}^{k} w_{li} = 1, \quad 1 \le i \le n, \quad \text{and} \quad w_{li} \in \{0,1\}, \quad 1 \le l \le k, \tag{2}$$

where $k(\le n)$ is a known number of clusters, $W = [w_{li}]$ is a k-by-n real matrix, $Z = [Z_1, Z_2, \ldots, Z_k] \in \mathcal{R}^{mk}$, and $d_c(Z_l, X_i)(\ge 0)$ is the simple matching dissimilarity measure between Z_l and X_i defined as

$$d_c(Z_l, X_i) \equiv \sum_{j=1}^{m} \delta(z_{lj}, x_{ij},) \tag{3}$$

where

$$\delta(z_{lj}, x_{ij}) = \begin{cases} 0, & x_{ij} = z_{lj} \\ 1, & x_{ij} \ne z_{lj} \end{cases}$$

Here, Z represents a set of k modes for k clusters[1]. It is easy to verify that the function d_c defines a metric space on the set of categorical objects.

Function (1) can be optimized with the k-means type algorithm **Algorithm 1**.

Algorithm 1 The k-means type algorithm

1. Choose an initial point $Z^{(1)} \in \mathcal{R}^{mk}$. Determine $W^{(1)}$ such that $J(W, Z^{(1)})$ is minimized.
   Set $t = 1$.
2. Determine $Z^{(t+1)}$ such that $J(W^{(t)}, Z^{(t+1)})$ is minimized.
   If $J(W^{(t)}, Z^{(t+1)}) = J(W^{(t)}, Z^{(t)})$, then stop; otherwise goto step 3.
3. Determine $W^{(t+1)}$ such that $J(W^{(t+1)}, Z^{(t+1)})$ is minimized.
   If $J(W^{(t+1)}, Z^{(t+1)}) = J(W^{(t)}, Z^{(t+1)})$, then stop; otherwise set $t = t + 1$ and goto Step 2.

In **Algorithm 1**, $W^{(t)}$ is updated using Function (3) in each iteration. After each iteration, $Z^{(t)}$ is updated using the method as shown as in Theorem 1. This guarantees the convergence of the algorithm.

Theorem 1 [The k-modes update method] *Let $\mathcal{X}$ be a set of categorical objects described by categorical attributes $A_1, A_2, \ldots, A_m$ and $DOM(A_j) = \{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(n_j)}\}$, where n_j is the number of categories of attribute A_j for $1 \le j \le m$. Let the cluster centres Z_l be represented by $[z_{l,1}, z_{l,2}, \cdots, z_{l,m}]$ for $1 \le l \le k$. Then the quantity $\sum_{l=1}^{k}\sum_{i=1}^{n} w_{li} d_c(Z_l, X_i)$ is minimized iff $z_{l,j} = a_j^{(r)} \in DOM(A_j)$ where*

$$\begin{aligned} & \left|\{w_{li} | x_{i,j} = a_j^{(r)}, w_{lj} = 1\}\right| \\ \ge & \left|\{w_{li} | x_{i,j} = a_j^{(t)}, w_{li} = 1\}\right|, \quad 1 \le t \le n_j, \end{aligned} \tag{4}$$

for $1 \le j \le m$. Here $|\mathcal{X}|$ denotes the number of elements in the set $\mathcal{X}$.

A proof of the theorem is given in [7]. According to (4), the category of attribute A_j of the cluster mode Z_l is determined by the mode of categories of attribute A_j in the set of objects belonging to cluster l.

4 Transition Frequency and Probability Matrices for Cluster Validation

A difficult task in cluster analysis is cluster validation, i.e., evaluation of clusters generated from a real data set by

[1] The mode for a set of categorical objects $\{X_1, X_2, \ldots, X_n\}$ is defined as an object Z that minimizes $\sum_{i=1}^{n} d_c(X_i, Z)$ [6].

a clustering algorithm. The task becomes more challenging in validating session clusters because of the order of variables. In this work, we attempted to use the transition frequency (or probability) matrix to validate session clusters. The rationale of using this technique is explained below.

Assume that a sequence of pages visited in a user session was generated by a "Markov process" of a finite number of states, (see [19]). The next page (state) to visit depends on the current page (state) only. Let n be the number of different pages. The user is said to be in the state $i(i = 1, 2, \ldots, n)$ if his current page is $P_i(i = 1, 2, \ldots, n)$. Let Q_{ij} be the probability of visiting page P_j, when the current page is P_i, i.e., the one-step transition probability. Q_{ij} and the transition frequency can be estimated by using the information of the sequence.

Suppose the transition probability matrix Q is known for a given user and X_m is the probability row vector of the user's state at his mth visit. We have

$$X_{m+1} = X_m Q \quad \text{and} \quad X_{m+1} = X_0 Q^m. \qquad (5)$$

The distribution of the transition frequency of the pages in the sequence (assuming the length of the sequence is much longer than the number of states n) should be consistent with the steady state probability distribution X in theory. This provides a method for verifying our assumptions. This Markovian approach can be illustrated by the following examples.

Consider the two sequences with three ($n = 3$) possible pages to visit:

$$A_1 = \underbrace{P_1P_2P_3}_{I}\underbrace{P_2P_2P_3}_{II}\underbrace{P_3P_2P_3}_{III}\underbrace{P_1P_2P_3}_{IV}$$

and

$$A_2 = \underbrace{P_2P_2P_3}_{II}\underbrace{P_1P_2P_3}_{I}\underbrace{P_1P_2P_3}_{IV}\underbrace{P_3P_2P_3}_{III}.$$

The sequence A_2 is obtained by interchanging the subsequences I and II and also the subsequences III and IV in the sequence A_1. Let $N^{(1)}(A_k)$ be the 3×3 one-step transition frequency matrix with the ijth entry $[N^{(1)}(A_k)]_{ij}$ being the number of transitions from page P_i to page P_j in the sequence A_k. Therefore we have

$$N^{(1)}(A_1) = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 4 \\ 1 & 2 & 1 \end{pmatrix}$$

and

$$N^{(1)}(A_2) = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 4 \\ 2 & 1 & 1 \end{pmatrix}.$$

One possible way to compare (distance) these two sequences is to consider the Frobenius norm of the difference of their transition frequency matrices, i.e. $||N^{(1)}(A_1) - N^{(1)}(A_2)||_F$, where

$$||B||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}[B_{ij}]^2}. \qquad (6)$$

Clearly this method works for two sequences of similar length. If we have a much shorter sequence $A_3 = P_1P_2P_3P_2P_2P_3$ which is the first half of the sequence A_1 then we have

$$N^{(1)}(A_3) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 2 \\ 0 & 1 & 0 \end{pmatrix}.$$

$\lVert N^{(1)}(A_1) - N^{(1)}(A_2)\rVert_F = \sqrt{2} = 1.414$
$\lVert N^{(1)}(A_1) - N^{(1)}(A_3)\rVert_F = \sqrt{8} = 2.828$
$\lVert N^{(1)}(A_2) - N^{(1)}(A_3)\rVert_F = \sqrt{10} = 3.162$

Table 1. Distances between sequences based on transition frequency.

Table 1 shows the distances among the sequences. We note that although A_3 is part of A_1, $||N^{(1)}(A_1) - N^{(1)}(A_3)||_F$ is quite large.

For two sequences of very different length, one may consider the one-step transition probability matrix instead of the transition frequency matrix. The one-step transition probability matrix can be obtained from the transition frequency matrix by dividing the entries of each row by its corresponding row sum. Denote the transition probability matrix of $N^{(1)}(A_k)$ by $Q^{(1)}(A_k)$, we have

$$Q^{(1)}(A_1) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{5} & \frac{4}{5} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix},$$

$$Q^{(1)}(A_2) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{5} & \frac{4}{5} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

and

$$Q^{(1)}(A_3) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} \\ 0 & 1 & 0 \end{pmatrix}.$$

The new distances under the one-step transition probability matrix approach are given in Table 2.

In the next section, we apply the transition frequency and probability matrix to analyze the clustering results for the real data set.

$\|\|Q^{(1)}(A_1) - Q^{(1)}(A_2)\|\|_F = \sqrt{\frac{1}{8}} = 0.354$
$\|\|Q^{(1)}(A_1) - Q^{(1)}(A_3)\|\|_F = \sqrt{\frac{1478}{3600}} = 0.632$
$\|\|Q^{(1)}(A_2) - Q^{(1)}(A_3)\|\|_F = \sqrt{\frac{3278}{3600}} = 0.954$

Table 2. Distances between sequences based on transition probability.

5 Experiments

Experiments were conducted on two real Web log files taken from the Internet. We first implemented a data preprocessing program to extract sessions from the log files and convert them into the cube model. We then extracted sessions with only Page ID values and used the k-modes algorithm to cluster these sessions. We used the transition probability matrix to analyze the validity of some identified clusters. Some of our analysis results are presented in this section.

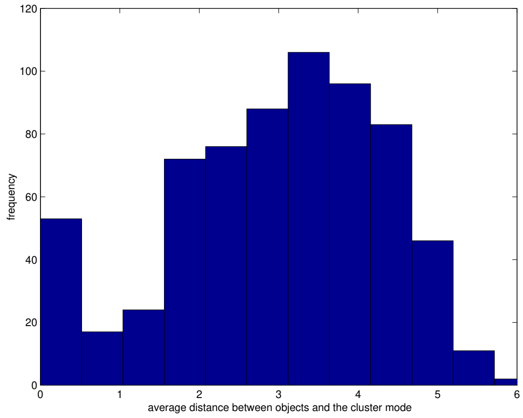
5.1 Web log files and preprocessing

We downloaded two Web log files from the Internet. The first data set was a Web log file from the NASA Kennedy Space Center WWW server in Florida. The log contained 1569898 transactions generated in the period of August 4-31, 1995. The second data set was a Web log file from the EPA WWW server located at Research Triangle Park, NC. This log contained 47748 transactions generated in 24 hours from 23:53:25 EDT, August 29, to 23:53:07, August 30, 1995.

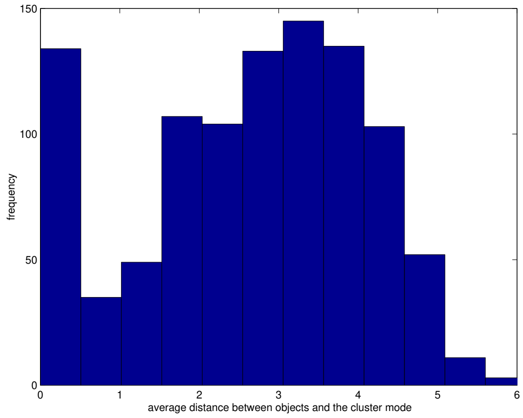
In preprocessing, we removed all the invalid requests and the requests for images. We used Host id to identify visitors and a 30 minutes time threshold to identify sessions. 120406 sessions were identified from the NASA log file and 2682 sessions were identified from the EPA log file. To simplify the cluster analysis, we filtered out the sessions with being less than 6 pages and more than 20 pages. From the rest sessions of the two log files, we generated six data sets with session lengths between 6 and 9, between 10 and 15, and between 16 and 20. The distributions of sessions from the log files are shown in Tables 3 and 4.

5.2 Cluster Analysis

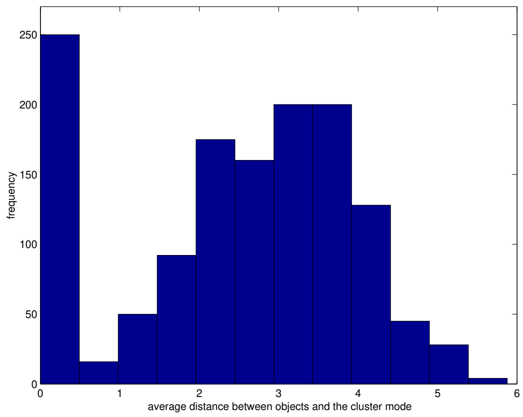
The k-modes algorithm is a variant of the popular k-means algorithm with a capability of clustering categorical data. To use the algorithm to cluster a data set, the first task is to specify a k, the number of clusters to create. However, k is generally unknown for real data. A heuristic approach is often based on the assumption that a few clusters



(a)



(b)



(c)

Figure 2. Distribution of average distances within clusters (a) when there are 674 clusters, (b) when there are 1010 clusters, and (c) when there are 1348 clusters for the data set NASA6-9.

Data set	NASA6-9	NASA10-15	NASA16-20
Session length	between 6 and 9	between 10 and 15	between 16 and 20
Number of sessions	13475	6492	4510

Table 3. Distribution of sessions in the NASA Web log data.

Data set	EPA6-9	EPA10-15	EPA16-20
Session length	between 6 and 9	between 10 and 15	between 16 and 20
Number of sessions	364	314	433

Table 4. Distribution of sessions in the EPA Web log data.

(say $k \leq 10$) exist in the data set. Although this approach works in many cases in practice, it has problems in clustering categorical session data. By exploring the session data from the two Web log files, we have observed that a cluster with a large number of similar sessions rarely exist. This is because in a complex Web site with variety of pages, and many paths and links, one should not expect that in a given time period, a large number of visitors follow only a few paths. If this was true, it would mean that the structure and content of the Web site had a serious problem because only a few pages and paths were interested by the visitors. In fact, most Web site designers expect that the majority of their pages, if not every one, are visited and paths followed (equally) frequently. In reality, some paths were followed more frequently than others in certain time period. Therefore, the session data should contain a fairly large number of small clusters.

Having the above observations, we tested to specify k as 5%, 10% and 15% of the total sessions in a data set. Even though we analyzed all six data sets, we only present the results of NASA6-9 here. The corresponding clusters for this data set are 674, 1010 and 1348, respectively. Although k was big in these specifications, k-modes can still process large data sets efficiently (the running time took a few seconds), which makes it a good candidate for this kind of mining tasks.

After generating the clusters, we need to identify which clusters are likely to present interesting session patterns. Because the large number of clusters, individual investigation of every cluster became difficult. We selected the average distance of objects to the cluster center, i.e., the

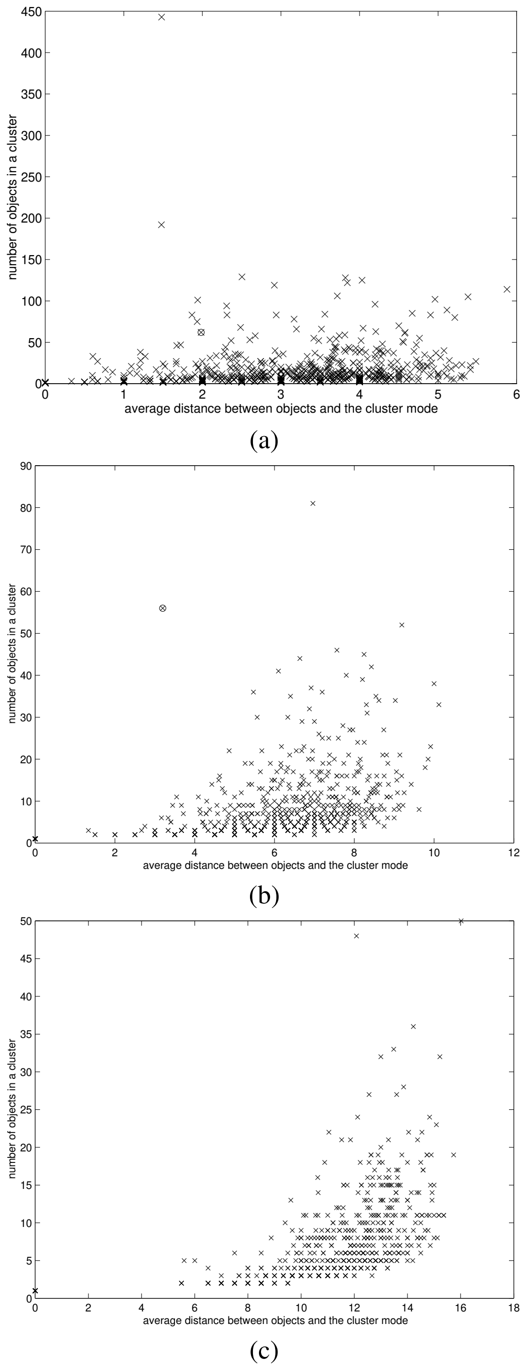


Figure 3. Distribution of numbers of objects and average distances within clusters (a) when there are 1348 clusters for the data set NASA6-9, (b) when there are 974 clusters for the data set NASA10-15, and (c) when there are 677 clusters for the data set NASA16-20.

mode of the cluster, as a measure for potential interesting clusters because the average distance implies the compactness of a cluster which is one of the important factors in cluster validation [8]. The distance was calculated using Equation (3) in Section 3. We expect that interesting patterns exist in compact clusters which have small average distances. Figure 2 shows the distributions of the number of clusters against the average distance of objects to cluster centers. From these figures, we can see that more clusters were created, more clusters with smaller average distances.

Next, we looked at the size of clusters, i.e., the number of sessions in a cluster. As we mentioned before, the chance of obtaining a large and compact cluster is very small. However, a reasonable size of clusters can represent the significance of interesting cluster patterns. In doing so, we plotted all clusters against their average distance and number of objects in each cluster. Figures 3 and 4 show the plots for the six data sets listed in tables 1 and 2. From these plots, we were able to identify the clusters with potential interesting patterns, given two thresholds of minimum size and maximum average distance. Tables 5 and 6 show potential interesting clusters identified from two data sets using this approach. For example, 34 clusters satisfy the conditions of (Average-Distance $\leq$ 2) and (Number-of-Sessions $>$ 10). These clusters are located in the upper left part of Figure 3(a). Because the average distances of these clusters are small, all of them present a clear pattern of Web site paths. Table 7 shows the pattern of the cluster that is marked as $\otimes$ in Figure 3(a).

These strong patterned clusters only cover 2.52% of the total clusters generated from the data set. To identify more potential clusters, we can ease the threshold of the average distance. For example, in the area of (2 $<$ Average-Distance $\leq$ 4), 186 clusters were identified from the data set NASA6-9, for instance, one of the interesting clusters with strong patterns is listed in Table 5. However, this was a grey area where not every cluster had a strong pattern. To select potential interesting clusters from the grey area, we used another condition that limited the clusters in which at least 50% of sessions had distances to the cluster center smaller than 2. Under this condition, 77 clusters were identified. Easing this condition could further identify more potential clusters from the grey area, for instance, see the last row in Table 5.

When the sessions became long, less potential interesting clusters could be identified. This was because the diversity of the sessions increased and similarity decreased. This trend can be observed from Figures 3(b) and 3(c) as well as Table 6. However, interesting clusters with strong patterns could still be found. The cluster $\otimes$ in Figure 3(b) was clearly singled out from other clusters. Its pattern is shown in Table 8. The mode of this cluster is $\{$59, 45, 47, 60, 76, 59, 45, 47, 60, 76, 0, 0, 0, 0, 0 $\}$. We

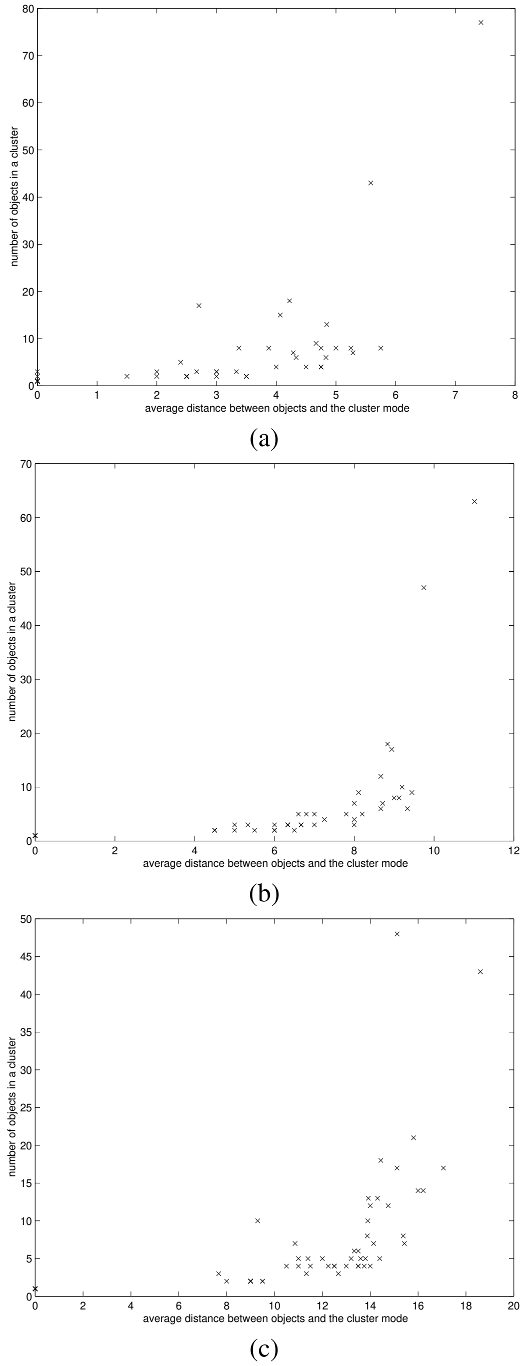


Figure 4. Distribution of numbers of objects and average distances within clusters (a) when there are 55 clusters for the data set EPA6-9, (b) when there are 47 clusters for the data set EPA10-15, and (c) when there are 65 clusters for the data set EPA16-20.

see that this mode constructs from two repeated pattern $\{59, 45, 47, 60, 76\}$. This patterns refers to { 7367 /images/ ; 12128 /icons/menu.xbm ; 12047 /icons/blank.xbm ; 10300 /icons/image.xbm ; 4782 /icons/unknown.xbm }. People are interested in finding some images from the NASA Kennedy Space Center WWW sever. Moreover, from the table, one can see a clear similarity among these sessions. Another interesting observation was that if there were not enough sessions in the data set, interesting clusters may not only show up in the short sessions. One can see this phenomenon in s 4. This was still caused by the diversity of long sessions. This exercise may tell us that a large number of sessions may be a precondition to mine interesting session patterns. If so, the efficiency of the clustering algorithm would become crucial in session data mining.

average distance within cluster ≤ 2 and the number of sessions > 10	34 (2.52%)
$2 <$ average distance within cluster ≤ 4 and the number of sessions > 10	186 (13.8%)
$2 <$ average distance within cluster ≤ 4, the number of sessions > 10 and at least 50% of these sessions have distance ≤ 2	77 (5.71%)
$2 <$ average distance within cluster ≤ 4, the number of sessions > 10 and at least 50% of these sessions have distance ≤ 3	127 (9.42%)

Table 5. Summary of the clustering results for the data set NASA6-9.

average distance within cluster ≤ 2 and the number of sessions > 10	0 (0.00%)
$2 <$ average distance within cluster ≤ 6 and the number of sessions > 10	29 (2.98%)
$2 <$ average distance within cluster ≤ 6, the number of sessions > 10 at least 50% of these sessions have distance ≤ 2	and 3 (0.31%)
$2 <$ average distance within cluster ≤ 4, the number of sessions > 10 at least 50% of these sessions have distance ≤ 3	and 7 (0.72%)

Table 6. Summary of the clustering results for the data set NASA10-15.

6 Cluster Validation

In section 4, we have proposed to use transition frequency matrix to validate session clusters because of the order of variables. For the cluster listed in Table 7, we computed the Frobenius norm of the differences between the objects in Table 7 and their mode. The results are listed in Figure 5. The Frobenius norm of the differences between the objects and their mode are between 0 and 3. The average difference is about 1.8. It is obvious that the cluster is valid.

In the data set NASA10-15, we found that there is a cluster containing the following session:

$$s = \{1, 59, 45, 47, 60, 76, 59, 47, 45, 60, 76, 1, 0, 0, 0\}.$$

This cluster contains 45 sessions. Their patterns are quite different and therefore the average difference within this cluster is about 8.2444. It is clear that this cluster is not valid.

However, the highlighted session looks quite similar to the mode of the cluster:

$$m = \{59, 45, 47, 60, 76, 59, 45, 47, 60, 76, 0, 0, 0, 0, 0\}$$

listed in Table 8. We note that their categorical distance measure $d_c(s, m)$ is equal to 11. Because of the large distance measure, we put the session s into a different cluster. Because of the order of variables in the session and the mode, we can consider the transition frequency matrix to measure the distance between s and m. According to the discussion in Section 4, the transition frequency matrix of the session s is given by

	1	45	47	59	60	76
1	0	0	0	1	0	0
45	0	0	1	0	1	0
47	0	1	0	0	1	0
59	0	1	1	0	0	0
60	0	0	0	0	0	2
76	1	0	0	1	0	0

The transition frequency matrix of the mode m is given by

	1	45	47	59	60	76
1	0	0	0	0	0	0
45	0	0	1	0	1	0
47	0	1	0	0	1	0
59	0	1	1	0	0	0
60	0	0	0	0	0	2
76	0	0	0	1	0	0

The Frobenius norm of the difference of these two transition frequency matrices is $\sqrt{2}$ which is less than 1.8. It is obvious that the session s should be in this cluster listed in Table 8.

Using the transition probabilty matrix, we can analyze the clusters of different lengths and validate clustering results more effectively. For instance, we pick a cluster mode

$$\begin{aligned} m' &= \{59, 45, 47, 60, 76, 59, 45, 47, 60, 76, 1, \\ &\quad 76, 59, 45, 47, 60, 0, 0, 0, 0\} \end{aligned}$$

from the data set NASA16-20. Because the length of the sessions in this data set is longer than 15, the clustering results for the data set NASA16-20 are not mixed with those

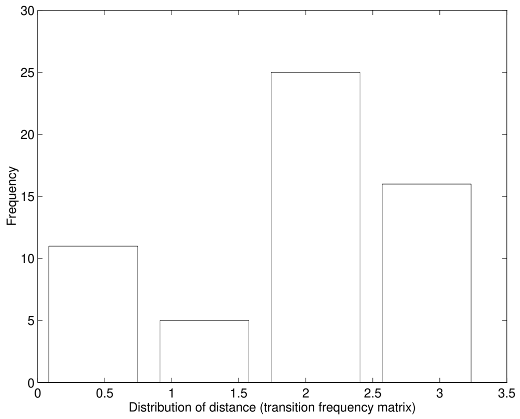


Figure 5. Distribution of distances within clusters listed in Table 6.

for the data set NASA10-15. However, we can analyze the transition probability matrices of m and m', so we can validate two clustering results. The transition probability matrix of the mode m' is given by

	1	45	47	59	60	76
1	0	0	0	0	0	1
45	0	0	1	0	0	0
47	0	0	0	0	1	0
59	0	1	0	0	0	0
60	0	0	0	0	0	1
76	$\frac{1}{3}$	0	0	$\frac{2}{3}$	0	0

The transition probability matrix of the mode m is given by

	1	45	47	59	60	76
1	0	0	0	0	0	0
45	0	0	0.5	0	0.5	0
47	0	0.5	0	0	0.5	0
59	0	0.5	0.5	0	0	0
60	0	0	0	0	0	1
76	0	0	0	1	0	0

The Frobenius norm of the difference of these two transition probability matrices is 1.64 which is not large. We can interpret both clusters should contains some common interesting patterns.

7 Conclusions

In this paper, we have presented the cube model to represent Web access sessions. This model is different from other cube approaches [11] [22] in that it explicitly identifies the Web access sessions, maintains the order of session's components (or Web pages) and uses multiple attributes to describe the Web pages visited in sessions. The three dimensional cube structure simplifies the representation of sequential session data and allows different data analyzes to be easily conducted, such as summary statistical analysis, clustering and sequential association analysis.

We have experimented the use of the k-modes algorithm to cluster categorical sessions extracted from two Web log files. Our preliminary cluster analysis has resulted in the following interesting observations:

1. Clusters with strong path patterns usually do not contain a large number of sessions due to the complexity of the Web structures and diversity of visitor's interests.

2. To effectively mine interesting path patterns using clustering techniques, the number of sessions in the data set should be sufficiently large. This requires that the clustering algorithm have to be efficient.

3. Because it is required to create a fair large number of clusters in each run of the clustering algorithm over a given data set, effective cluster evaluation methods are needed to identify potential interesting clusters with strong path patterns.

We have tested using the average distance of sessions to the cluster center and the size of clusters as two criteria to select potential interesting clusters from the set of clusters generated by the k-modes algorithm. Our results have shown that those criteria were effective. We were able to identify clusters with strong path patterns. Furthermore, we have proposed and tested the transition frequency (or probability) matrix approach to validating clusters of sessions. Our initial results have shown that this approach is promising in cluster validation of sequential data but more studies are needed.

In our future work we will conduct cluster analysis on sessions with more attributes such as time and category. For example, if two sessions have a similar set of pages, whether the time spent on each would make them different. If we consolidate Web pages into categories with a classification scheme, what kind of cluster patterns would result? How the cluster patterns are related to the topology of the Web site? Can these patterns be used to improve the Web site structure? How can the Web topology be used as constraints in the clustering algorithm? All these interesting questions need further studies to answer. Moreover, we will conduct clustering analysis using transition frequency (or probability) matrix to cluster sessions.

References

[1] Chen, M. S., Park, J. S. and Yu, P. S. (1998) Efficient data mining for path traversal patterns. IEEE Trans-

actions on Knowledge and Data Engineering, Vol. 10, No. 2, pp. 209-221.

[2] Cooley, R., Mobasher, B. and Srivastava, J. (1999) Data preparation for mining World Wide Web browsing patterns. Knowledge and Information Systems, Vol. 1, No. 1, pp. 1-27.

[3] Etzioni, O. (1996) The World Wide Web: quagmire or gold mine? Communications of the ACM, Vol. 39, No. 11, pp. 65-68.

[4] Fu, Y., Sandhu, K. and Shih, M. (1999) Clustering of Web users based on access patterns. WEBKDD99, Springer.

[5] Han, J., Cai, Y. and Cercone, N. (1992) Knowledge discovery in databases: an attribute-oriented approach. In Proceeds of VLDB92, Canada.

[6] Huang, Z. (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery, Vol. 2, No. 3, pp. 283-304.

[7] Huang, Z. and Ng, M. K. (1999) A Fuzzy k-modes algorithm for clustering categorical data. IEEE Transactions on Fuzzy Systems, Vol. 7, No. 4, pp.446-452.

[8] Jain, A. K. and Dubes, R. C. (1988) Algorithms for Clustering Data. Prentice Hall.

[9] Joshi, A. and Joshi K. (1999) On mining Web access logs. Technical Report, CSEE Department, UMBC, MD, USA. http://www.cs.ubmc.edu/ joshi/webmine/publications.html

[10] Kamdar, T. and Joshi, A. (2000) On creating adaptive Web servers using weblog mining. Technical report CS-TR-00-05, CSEE, UMBC, USA. http://www.cs.ubmc.edu/ joshi/webmine/publications.html

[11] Kimball, R. and Merx, R. (2000) The Data Webhouse Toolkit – Building Web-Enabled Data Warehouse. Wiley Computer Publishing.

[12] Kosala, R. and Blockeel, H. (2000) Web mining research: a survey. SIDKDD Explorations, Vol. 2, No. 1, pp. 1-15.

[13] Magid, J., Matthews, R. D. and Jones, P. (1995) The Web Server Book – Tools & Techniques for Building Your Own Internet Information Site. Ventana Press.

[14] Nasraoui, O., Frigui, H., Joshi, A. and Krishnapuram, R. (1999) Mining Web access logs using relational competitive fuzzy clustering. Proceedings of the Eight International Fuzzy Systems Association Congress - IFSA99.

[15] Ng, R. and Han, J. (1994) Efficient and effective clustering methods for spatial data mining. In Proceedings of VLDB, 1994.

[16] Shahabi, C., Faisal, A., Kashani, F. B. and Faruque, J. (2000) INSITE: A tool for real-time knowledge discovery from users Web navigation. Proceedings of VLDB2000, Cairo, Egypt.

[17] Spiliopoulou, M. and Faulstich, L. C. (1998) WUM: A Web utilization miner. In EDBT Workshop WebDB98, Valencia, Spain, Springer.

[18] Srivastava, J., Cooley, R., Deshpande, M. and Tan, P. (2000) Web usage mining: discovery and applications of usage patterns from Web data. SIGKDD Explorations, Vol. 1, No. 2, pp. 12-23.

[19] Taha, T. (1991) Operations Research, 3rd Edition, Collier Macmillan, N.Y., U.S.A.

[20] www.w3.org/Daemon/User/Config/Logging.thml

[21] W3C (1999) Web Characterization Terminology & Definitions Sheet. W3C Working Draft 24-May, 1999. http://www.w3.org/1999/05/WCA-terms/.

[22] Zaiane, O. R., Xin, M. and Han, J. (1998) Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs. Proceedings of Advances in Digital Libraries Conference (ADL'98), Santa Barbara, CA, April 1998, pp.19-29.

[23] Zhang, T. and Ramakrishnan, R. (1997) BIRCH: A new data clustering algorithm and its applications. Data Mining and Knowledge Discovery, Vol. 1, No. 2, pp. 141-182.

Object Attributes								
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	1	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	23	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	59	0	0	0
59	47	60	45	76	1	161	0	0
59	47	60	45	76	1	25	0	0
59	47	60	45	76	1	79	0	0
59	47	60	45	76	13	0	0	0
59	47	60	45	76	23	0	0	0
59	47	60	76	45	1	0	0	0
59	47	60	76	45	1	0	0	0
59	47	60	76	45	1	0	0	0
59	47	60	45	76	1	86	179	0
59	47	60	76	45	1	0	0	0
269	47	60	45	266	1	0	0	0
59	47	60	45	76	28	1	0	0
59	47	60	45	76	3	131	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	45	60	47	76	1	0	0	0
59	47	60	45	1	76	0	0	0
59	47	60	45	59	76	0	0	0
59	47	60	45	59	76	0	0	0
59	47	76	45	60	161	0	0	0
59	47	76	45	60	45	0	0	0
59	47	76	45	60	59	0	0	0
59	60	47	45	76	59	0	0	0
59	60	47	45	76	59	0	0	0
59	60	76	45	47	1	0	0	0
59	60	76	45	47	1	0	0	0
59	60	76	45	47	1	0	0	0
59	76	60	45	47	99	0	0	0
59	47	60	45	76	28	25	213	0
59	47	60	45	76	28	6	19	0
59	47	60	45	76	3	17	232	0
59	47	60	45	76	3	6	843	0
59	47	60	45	76	59	47	45	0
59	47	60	45	76	99	46	101	0
269	47	60	45	46	48	0	0	0
275	47	60	45	46	30	0	0	0
59	45	60	47	76	1	206	0	0
59	45	60	47	76	1	48	0	0
59	45	60	47	76	1	59	0	0
59	47	60	45	59	60	47	45	0
60	76	47	45	59	1	0	0	0
59	60	47	45	76	99	46	0	0
59	47	60	45	76	99	46	101	46

Table 7. The cluster $\otimes$ in Figure 3(a) where 0 represents the missing page. The mode of this cluster is $\{59,47,60,45,76,1,0,0,0\}$.

Object Attributes														
59	47	60	45	76	59	45	47	60	76	1	0	0	0	0
59	45	47	60	76	59	45	60	47	76	0	0	0	0	0
59	45	47	60	76	59	47	45	60	76	0	0	0	0	0
59	45	47	60	76	59	47	45	60	76	0	0	0	0	0
59	47	76	45	60	59	45	47	60	76	0	0	0	0	0
59	47	45	60	76	59	47	45	60	76	0	0	0	0	0
59	47	45	60	76	59	47	45	60	76	0	0	0	0	0
59	47	60	76	45	59	45	47	76	60	0	0	0	0	0
59	47	60	45	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	47	23	92	74	30	33	224	0	0
59	45	47	60	76	59	45	47	60	76	59	0	0	0	0
59	45	47	60	76	59	45	47	60	76	45	60	79	157	79
59	45	47	60	76	59	1	17	23	86	0	0	0	0	0
59	45	47	76	60	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	770	59	45	47	60
59	47	45	60	76	59	47	45	60	76	36	0	0	0	0
59	45	47	60	76	59	47	45	60	76	59	47	45	60	76
59	76	45	47	60	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	60	47	76	59	45	47	60	76	0	0	0	0	0
45	47	59	60	76	59	45	60	47	76	59	0	0	0	0
59	45	60	47	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	59	0	0	0	0
59	45	60	47	76	59	45	47	60	76	0	0	0	0	0
59	45	60	47	76	59	45	60	76	47	0	0	0	0	0
59	47	60	76	45	59	45	47	60	76	770	0	0	0	0
59	47	45	60	76	59	47	45	60	76	59	47	45	60	76
59	45	47	60	76	59	47	60	76	45	59	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	60	47	45	76	59	45	60	47	76	0	0	0	0	0
59	45	47	60	76	59	1	3	374	294	46	345	80	0	0
59	45	47	60	76	59	1	9	11	21	53	49	142	0	0
59	45	47	60	76	59	45	47	60	76	1	0	0	0	0
59	45	47	60	76	59	45	47	60	76	1	25	0	0	0
266	45	47	1	48	266	45	47	1	266	269	46	60	0	0
59	45	47	60	76	59	45	47	60	76	63	17	0	0	0
59	45	47	60	76	59	45	47	60	76	60	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	47	45	60	76	59	45	47	60	76	0	0	0	0	0
59	47	45	60	76	59	45	47	60	76	0	0	0	0	0
59	47	45	60	76	59	45	47	60	76	0	0	0	0	0
59	60	45	76	47	59	45	47	60	76	59	0	0	0	0
59	47	45	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	47	60	45	76	99	46	0	0	0
59	47	45	60	76	59	47	45	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	45	47	60	76	59	45	47	60	76	0	0	0	0	0
59	47	45	60	76	59	47	45	60	76	0	0	0	0	0
59	60	45	76	1	59	45	47	60	76	1	50	0	0	0
59	45	47	76	59	60	59	47	60	76	45	0	0	0	0
59	45	47	60	76	59	45	47	60	76	9	0	0	0	0

Table 8. The cluster $\otimes$ in Figure 3(b), where 0 represents the empty page. The mode of this cluster is $\{59, 45, 47, 60, 76, 59, 45, 47, 60, 76, 0, 0, 0, 0, 0\}$.