Cut-and-Pick Transactions for Proxy Log Mining

Wenwu Lou, Guimei Liu, Hongjun Lu, and Qiang Yang

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{wwlou,cslgm,luhj,qyang}@cs.ust.hk

Abstract. Web logs collected by proxy servers, referred to as *proxy logs* or *proxy traces*, contain information about Web document accesses by many users against many Web sites. This *"many-to-many"* characteristic poses a challenge to Web log mining techniques due to the difficulty in identifying individual access transactions. This is because in a proxy log, user transactions are not clearly bounded and are sometimes interleaved with each other as well as with noise. Most previous work has used simplistic measures such as a fixed time interval as a determination method for the transaction boundaries, and has not addressed the problem of interleaving and noisy transactions. In this paper, we show that this simplistic view can lead to poor performance in building models to predict future access patterns. We present a more advanced *cut-and-pick* method for determining the access transactions from proxy logs, by deciding on more reasonable transaction boundaries and by removing noisy accesses. Our method takes advantage of the user behavior that in most transactions, the same user typically visits multiple, related Web sites that form clusters. These clusters can be discovered by our algorithm based on the connectivity among Web sites. By using real-world proxy logs, we experimentally show that this *cut-and-pick* method can produce more accurate transactions that result in Web-access prediction models with higher accuracy.

1 Introduction

Web logs collected by Web servers or by proxy servers contain information about user accesses to Web documents. Web logs are usually very large in size, and the analysis of these logs calls for more advanced methods such as data mining techniques. Recently, the application of data mining techniques to the Web logs, referred to as *Web Log Mining*, has draw extensively attention from researchers from various disciplines [CSM97]. From Web logs, one can identify user access patterns which can be used for various purposes such as Web site design [SPF99], page recommendation [PPR96,DH99], Web caching and prefetching [PM96,PM98,SKS98,PP99,YZL01].

A key problem for Web log mining, e.g., association rule mining, is to identify individual user transactions in a Web log [CMS99]. When correctly extracted, a user transaction can be regarded as a semantically meaningful representation of a sequence of page references corresponding to a single search task of the user. Previous work [BL99,BL00,NM00] of Web log mining has been conducted on Web server logs, and typically used simplistic measures such as a fixed time window or a pre-defined time interval as a determination method for the transaction boundaries. These simple measures, however, may lead to poor performance in the context of proxy Web log mining. This is because in proxy logs, page accesses made by Web users against different servers are often interleaved with each other. This makes it more difficult to identify user transaction due to two reasons. The first reason is that in a proxy log, user transactions are often interleaved with each other. This is because a) there is no easy way to identify a real-world user in proxy logs, b) the same user may concurrently perform multiple search tasks. The second reason is that unlike the case of user transactions in server logs, user transactions in proxy logs often involve page accesses to multiple Web sites. In both cases, the simplistic approaches, such as a fixed time interval based method, may not perform well on proxy logs.

In this paper, we will explore a more advanced approach for transaction identification in Web proxy logs, by deciding on more reasonable boundaries between transactions and by selecting the right reference sequence in each transaction. We assume that in one transaction, the same user usually focus on one subject and they often visits one or multiple Web sites that are related to the subject. They focus on different subjects in different transactions. Web users' online browsing and search behavior is often threaded by the linkage information among the Web sites. A group of Web sites that users often visit together forms a cluster of sites, which can serve as a representative of a particular subject. These clusters can be discovered by our Web site clustering algorithm based on site traversal graphs constructed from the proxy logs. Our approach is to apply this common user behavior in transaction identification by identifying the subject of each transaction, so that its boundary and content can be determined more accurately. The idea is as the following: if two consecutive page references in a proxy log visit two Web sites that fall in two clusters, the two visits are regarded as irrelevant and are therefore classified into two user transactions. Our experiment results showed that, by also taking into account the client-side information and the time stamp of a page references, our *cut-and-pick* method can produce more realistic transactions that lead to better Web-access prediction models with higher accuracy.

Our study distinguishes from previous work in mainly two aspects. First, previous work has discussed the issue of identifying transaction boundaries, but no one has addressed the problem of content selection of transactions. Since in a proxy log, multiple transactions are interleaved with each other as well as with noise, even when it is filtered into independent sequences by IP addresses. Picking out the right sequence for each transaction becomes a tough task. Taking advantage of the common user behavior, our approach not only addresses the problem of transaction boundary identification, but also provides a solution to select page reference sequence for each user transaction. Second, most of the previous studies have been analyzing of the *server logs*, but not the proxy logs. A server log only contains information of pages accesses on a single Web server. Consequently, the problem of transaction identification is relatively simple as there's no need to identify transactions that involve multiple Web sites.

The remaining of this paper is organized as follows. Section 2 reviews related work. This is followed by a brief discussion of three transaction identification approaches we compared to in this study in Section 3. Section 4 describes the procedure of identifying Web site clusters from proxy logs. Our *cut-and-pick* approach for identification of user transactions is presented in Section 5. The effectiveness of our algorithm is evaluated in this paper by comparing the quality of derived prediction models trained on the user transactions identified by our algorithm and other approaches. The experimental results are reported in Section 6. We conclude our paper in Section 7 with a brief discussion of the future work.

2 Related Work

The Web log data records information of the Web users' visits to Web sites. Researchers have been trying to discover significant user patterns from the logs, and to build prediction models to make predictions on users' future access needs. These prediction models can be used to reduce the network traffic [SKS98,YZL01], to judge the quality of Web sites [SP01] and to automatically customize the user interface [ADW01,PE00].

Various data preparation techniques have be described in [CMS99]. Three transaction identification approaches are compared in this study. The reference length transaction identification approach is based on the assumption that the amount of time that a user spends on a page depends on how well this page is related to the user's search task. The second approach called maximal forwarding transaction identification, which was originally proposed by [CPY96]. It defines each transaction as a maximal set of page references in a path that contains no backward reference. While this approach is reasonable in dealing with server Web logs, it is not applicable for the transaction identification in proxy logs. This is because in a proxy log, there's no single "first" pages to start with. In addition, user sessions are normally much longer in a proxy log than on a server log. For example, it is often the case that a user visits several pages located at some related Web sites in a row, but stays on each site only over a limited number of pages. The third approach is the time window transaction identification approach. It assumes that meaningful transactions have an overall average length associated with them. It simply partitions a user session into time intervals no larger than a specified parameter.

A significant amount of work has been done in the sequential data mining area. Agrawal et al. [AMS$^+$96,AS95] discovered frequent items that are bought together in certain sequential order using intra-transaction and inter-transaction patterns. The general idea of sequential mining is an extension of the original Apriori algorithm, where k-large itemsets, with k denoting the size of itemsets, are recognized from k-candidates and are used to generate (k+1)-candidates. The resultant association rules correspond to sequences of Web accesses that comprise the LHS of a prediction rule.

Another line of work is to discover frequent patterns on users' browsing path that are longest in length; these are called the longest repeating subsequences. When the users are observed to make a sequence of Web page visits, these path patterns are consulted to determine the next page that the user might access. In fact, researchers have studied algorithms that discover longest paths [PP99] or a hybrid of N-gram models [SYZ00,PP99,BL99].

A unifying theme of the above work is that they all build their models based on Web server logs. These are the log files accumulated on the server side, rather than on the client or proxy side. When we deal with Web logs that come from the proxy side, new problems arise. For example, instead of accessing the Web pages on one server in each user session, a user may access multiple servers in one session. This many to many relationship provides both a challenge and a new opportunity for data mining researchers. In this paper, we address the new problems from the Web proxy-server logs.

3 Framework

In this section, we present the framework of our approach for identifying user transactions in proxy logs. For clarity of presentation, we first define some terminologies which we use frequently throughout this paper.

3.1 Definitions

In this paper, a *proxy log L* is viewed as a sequence of page references. Each reference R is defined as a quintuplet $R :< R_{id}, IP, S_{id}, P_{id}, t >$, where R_{id} represents for a unique identifier of a page reference within a Web log, IP for the requesting IP, S_{id} for the server, and P_{id} for the page being requested respectively. The t value indicates the time of the request. While the actual representation of a page reference may contain additional information, it is irrelevant to this paper and is excluded from the analysis.

We first define IP sequence. Intuitively, an IP sequence is the lifetime sequence of Web accesses originating from the same IP address in an entire Web log.

Definition 1 (IP sequence). *An IP sequence, denoted by L_{IP}, is an order-preserving subsequence of page references in a proxy log L, such that: a). $\forall R \in L_{IP}, R.IP = IP$; and b). $\forall R \in L, and R.IP = IP, R \in L_{IP}$;*

In other words, a page in a Web log belongs to an IP sequence if and only if it shares the same IP address. In related literature, an IP sequence is often referred to as a *user sessions* as they assume that each IP address represents an individual Web user. However, we note this is misleading in a proxy log, as it is often the case that multiple users may share the same IP address on a proxy, either due to multiple accesses of the same computer by a group of users or by dynamic allocation of an IP address to a set of client terminals. Moreover, in some cases such as a public computer in a shopping mall or library, the same computer is shared among multiple users. Thus, a single IP sequence can actually contain requests from multiple users. Deriving IP sequences from a proxy log is the first step that most Web log mining algorithms perform. This can be done by partitioning the Web log L according to U_{ip}.

An IP sequence is divided into segments of subsequences by time windows. The time windows represent significant "idle" periods. Each IP subsequence is called an IP transaction.

Definition 2 (IP transaction). *An IP transaction, denoted by T_{IP}, is an order-preserving subsequence of page references of an IP sequence L_{IP}, such that: a) if $R_i \rightarrow R_j$ are two consecutive references in T_{IP}, $R_i \rightarrow R_j$ are also two consecutive references in L_{IP}; b) Let $T^1_{IP}, T^2_{IP}, \ldots, T^n_{IP}$ be all IP transactions that derived from one IP sequence L_{IP}, then there exist a permutation $i_1, i_2, \ldots, i_n$ of $1, 2, \ldots, n$, such that $L_{IP} = T^{i_1}_{IP} + T^{i_2}_{IP} + \ldots + T^{i_n}_{IP}$.*

"+" above means string concatenation. Note that our notion of *IP transaction* is often referred to as *user transaction* in previous literature (e.g., [CMS99]). However, as we have shown that an IP sequence can actually contain requests from multiple users, an IP transaction may also be composed of page references from *multiple users*. Thus, to be more precise, we call it an *IP transaction*, and reserve the term *user transaction* to the following. The purpose of this study is to identify these *real* user transactions.

Definition 3 (user transaction). *A user transaction, denoted by UT_{IP}, is any order-preserving subsequence of page references of an IP sequence L_{IP}. Each user transaction only contains the page references that semantically correspond to an individual search task conducted from an IP address IP.*

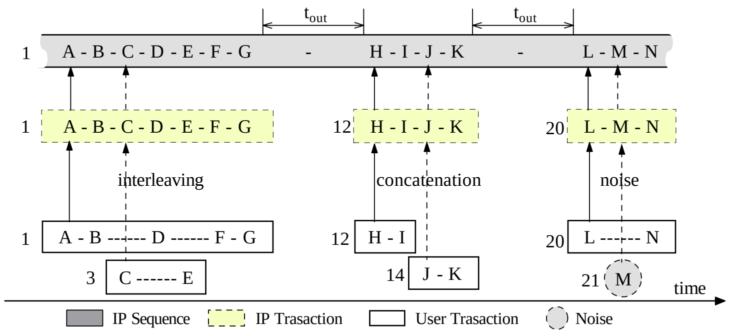


Fig. 1. IP Sequence, IP Transaction and User Transaction

Note that the definition is intentionally left vague because there is difficulty in identifying the semantically identical search tasks. However, our work reported in this paper will attempt to approximate this note empirically.

Page references in a *user transaction* are not necessarily consecutive in an IP sequence, but they must come from the same user's single search task. The relationship among IP sequences, IP transactions and user transactions is illustrated using an example in Figure 1. In this example, an IP client runs multiple user transactions in parallel. The final ordering of page references depends on the ordering of their arrival at the proxy. Three IP transactions can be identified by the t_{out} value. The first IP transaction is composed of two user transactions that are interleaved with each other. The second one is a concatenation of two consecutive user transactions. In the third IP transaction, it is a noisy access. Thus, in order to identify the *user transactions* correctly from proxy logs, not only the transaction boundaries need to be identified, but also the transaction contents, i.e., the sequence of pages references, need to be correctly selected.

3.2 Transaction Identification in Proxy Logs

The problem of transaction identification can be described as follows: given a proxy log L, derive a set of user transactions from the logs. Basically, there are two tasks. One is to find the "start" and "end" for each transaction, i.e. the transaction boundaries. In fact, we only need to find out the "ends" of transactions because the reference following the "end" of a transaction will be the "start" of the next transaction. The second task is to select the right subsequence of references for each transaction, by decomposing interleaved transactions and removing noisy accesses in the original sequence.

In the remaining of this section, we briefly describe the three transaction identification approaches that we studied in this paper, including a simplistic approach through fixed time interval, a new proposed server-based approach to identify transactions within individual Web sites, and an overview of our *cut-and-pick* approach.

Fixed Time Interval Approach. A popular solution applied in previous work has been a fixed time interval t_{out} approach, where users are assumed to start a new transaction when the time interval between two page references exceeds the specified t_{out} threshold. The rational behind this simplistic approach is that users typically make longer pauses between transactions than within a transaction. Hence, if we can derive a reasonable cutoff time value t_{out} from the data, we can identify transactions based on the time value.

Our procedure in implementing this approach is as the following: Scan the proxy log from the beginning, on encountering a page reference R from an new IP address, we always start a new transaction of this new IP address, taking R as its first page reference. If R is from an existing IP address with its last page reference being R_{last}, we check whether the time interval between R_{last} and R has exceeded a given threshold t_{out}. If $R.t - R_{last}.t < t_{out}$, we add the R to the end of its current transaction. Otherwise, we start a new transaction of this IP starting with R.

In essence, the transactions that are identified by this simplistic approach are IP transactions, but not user transactions. While it is easy to implement, this approach has a number of major drawbacks. First, it is not appropriate to assume that a user makes longer pauses between transactions than within a transaction. Secondly, selecting an appropriate t_{out} is critical to its performance and has to be determined on a case by case basis. Generally, it is determined by analyzing the density distribution of time interval between pages reference. Thirdly, it cannot identify transactions that are interleaved with each other and it can not remove noisy pages from the log. Here, a noisy page refers to a page that does not belong to any user transactions.

Server Based Approach. The server based approach is to first partition a proxy log into many server logs by server id S_{id}, then apply the similar idea of time interval approaches for processing each individual server log. In adopting this method, a basic assumption is that a user typically visits a single Web site within one transaction. While this approach is effective in identifying transactions that only involve one Web site, it fails to identify transactions that involve multiple Web sites.

Overview of *Cut-and-Pick.* As illustrated in Figure 1, the presence of interleaved transactions and noisy accesses calls for more sophisticated approaches for user transactions identification in Web logs. When dealing with proxy logs, an effective transaction identifier is one that is able to identify interleaved transactions and transactions with noise, and can capture both the intra-site transactions and the inter-site transactions. The *cut-and-pick* approach that we proposed in this paper tries to exploit available information to identify user transactions besides simple indicators like transaction length and time interval between page references. Note that each semantically meaningful user transaction has to be associated with a search task that has a unique purpose or subject. Thus, by identifying the subject of a current transaction, one can distinguish multiple interleaved transactions in a proxy log. We notice that most users browse the Web by following the links and typically visit one site or multiple, related Web sites within one transaction. These sites form clusters of related information. Thus, by grouping Web sites into clusters that users often visit together, the subject of a transaction can be roughly identified by the cluster information of Web sites visited within a transaction. This enables us to cut on the right boundaries among user transactions as well as to pick the right "content" for individual transactions.

In *cut-and-pick* approach, the Web site clusters are first identified offline in a data preprocess phase. We first extract site inter-referencing information from the proxy logs, then identify Web site clusters using our connectivity-based clustering algorithm. The *cut-and-pick* algorithm then combines the IP address, time stamp and the identified clusters for deciding on whether a current checked reference in a proxy log needs be treated as a subsequent reference of a current transactions or a starting page reference of a new transaction. The details of our clustering method is presented in the next two sections.

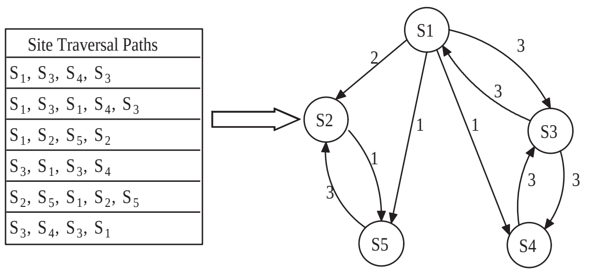
4 Identify Web Site Clusters

In this section, we describe how to identify Web site clusters without looking at the Web site contents. Web pages on one Web site often contain linkage information to multiple related information sources for facilitating Web users' locating process of interesting information. Typically, Web users visit multiple, related Web sites by following the links. This confines Web users' browse activities to a small portion of the Web and to certain patterns, which also makes it possible for us to discover the connection among Web site based on user access patterns, without looking at contents of Web sites.

4.1 Extract Inter-site Referencing Information

In the data preprocessing phase, a set of IP transactions are first derived from the log. As we are only interested in looking at the relationship between servers, the page information in the IP transaction can be excluded from the analysis. Therefore, when deriving the IP transactions, we omit the page information and only keep the sever sequences. We use an example to illustrate this process. Suppose an IP transaction contains the following page references, $(S_1.A, S_2.B, S_2.C, S_1.D, S_1.E, S_2.C)$, where S_i is a server I and A, B, C etc. are the pages located on these servers. By removing the page information, the resulting site access sequence looks like $(S_1, S_2, S_2, S_1, S_1, S_2)$. We than further abbreviate it to (S_1, S_2, S_1, S_2). The resulting transactions with only site information will be referred to as *site traversal paths.*

By doing so, some of the adjacent references in IP transaction will collapse into single references if multiple Web pages are from the same server. This helps reduce the average length of each IP transaction. In the extreme case, when all the references in an IP transaction collapse into one reference, this transaction will be removed from further analysis as it provides no information about links between Web sites. As such extreme cases are often observed for Web transactions, analyzing the references at server level can also greatly reduce the total number of transactions.



Site Traversal Paths
S_1, S_3, S_4, S_3
S_1, S_3, S_1, S_4, S_3
S_1, S_2, S_5, S_2
S_3, S_1, S_3, S_4
S_2, S_5, S_1, S_2, S_5
S_3, S_4, S_3, S_1

Fig. 2. Constructing STG from Site Traversal Pathes

Once we generate the site traversal path set, a *Site Traversal Graph* is then constructed, where each node in the graph represents a Web site and each direct edge represents a connection from its origin site to its end site. An illustration example for constructing STG is shown in Figure 2.

4.2 Measuring *Distance* between Web Sites

Based on the site traversal graph, we identify Web clusters by the strength of the connections among sites, i.e., whether two sites will be classified into one cluster depends on how strong the connection is between them. In our approach, the connectivity between two sites is measured by two parameters: support and confidence. Let $E :< S_i, S_j >$ be an edge from Web site S_i to S_j, the support of E, denoted by $freq(S_i, S_j)$, is defined as the frequency of inter-references between S_i to S_j. The confidence of E is defined as $\frac{freq(S_i,S_j)}{freq(S_i)}$.

The reason that we include confidence as an indicator of the connectivity among two sites is that the distribution of page references on the Web is highly skewed in the site dimension and using support as the single metric will result in a large number of sites to be glued together by some hot site erroneously. For example, in our data sets, while the average number of accesses to one site is as few as 4 times per day, a hot site can actually have as many as 20,000 accesses, which provides enough “support” for most sites to be linked to the hot site.

4.3 The Clustering Algorithm

Our clustering algorithm takes a traversal graph, a support threshold and confidence threshold as inputs. All edges with support or confidence value less than the corresponding threshold values will be removed from the graph. Then nodes in each connected sub-graph in the remaining site traversal graph constitute a site cluster. The detailed algorithm is described in figure 3.

```
ALGORITHM: ClusterSite($STG$, $minSup$, $minConf$)
01) $S \leftarrow$ {all sites};
02) $cluster \leftarrow 0$;
03) WHILE ($\exists$ edge $\varepsilon \in STG$ and $\varepsilon.sup < minSup$ and $\varepsilon < minConf$)
04)       Remove $\varepsilon$ from $STG$;
05) WHILE ($S$ is not empty) {
06)       $H \leftarrow$ the 1st site in $S$;
07)       $STG_H \leftarrow$ BFS($STG$, $H$);        /* $STG_H$ is a subgraph of $STG$  */
08)                                           /* that is reachable from site $H$  */
09)       $\forall\ s \in STG_H$) {
10)             $s.cluster \leftarrow cluster$;
11)             Remove site $s$ from $S$;
12)       }
12)       $cluster \leftarrow cluster + 1$;
13) }
```

Fig. 3. The Clustering Algorithm

5 Cut-and-Pick

Assuming that the site clustering information has been obtained by applying our clustering algorithm, we now describe how to use this information to obtain more accurate boundaries of user transactions.

Recall that the simplistic time interval approaches has assumed that an IP address uniquely identifies a user, and that a user always makes a longer pause between transactions than within a transaction. We have identified two exceptions to the simplistic case.

- First, in proxy logs, the real-world individual users cannot be uniquely identified. The only information about user identification in a proxy log is the information about client machines, e.g., IP addresses. Thus, while multiple users are mapped into one IP addresses, the transactions conducted by different users are mixed up together.
- Second, although one can assume that a long pause indicates a gap between two user transactions, the opposite is not always true. In other words, user may often start a new transaction immediately after one transaction. For example, a sport fan may too impatient to finish his current browsing but quickly change to view a football match on live at FOXSPORTS.COM. Moreover, it is also quite common for a user to conduct multiple transactions at the same time in parallel in a multi-task system.

We notice that Web site cluster information can be used for managing the above-mentioned difficult cases effectively. First, for the case that multiple users are mapped into the same IP addresses, Web site cluster information can helps to identify these different individuals in proxy logs, as different users have different interests and will presumably visit different cluster of Web sites. In fact, the probability is very low that two users visit the same cluster Web site simultaneously. Second, for the case that multiple transactions are interleaved or concatenate with each other, Web cluster information can help to decompose them into individual transactions, while the simple time interval based approach can not. This is because that different transactions have different focuses, and in one transaction, a user typically visit multiple, related Web sites that fall into one cluster.

Basically, the *cut-and-pick* approach integrates the three indicators for user transaction identification, including IP address, time interval, and Web cluster information. In outmost loop, page references are filtered into individual *IP sequences* according to the requesting IP addresses. Then, for each IP sequences, there are three cases to determine. Let R be a current page reference, P be the previous page reference in the same IP sequences of R. If $R.time - P.time > t_{out}$, i.e., the time interval between the current page reference and a previous one exceeds the time interval threshold t_{out}, it is considered that R starts a new transaction. Otherwise, we check for the site information of R and P. If they fall into one cluster, R is regarded as a subsequent visit of P in the same transaction; otherwise, R and P are considered to be irrelevant, but happen to arrive at the proxy one after another.

The detailed *cut-and-pick* procedure is provided in Figure 4. For each IP, we maintain an index on its current active user transactions. Active user transactions refer to those transactions that haven't been completely identified when we scan the IP sequence. A subtle problem is that we need to check the time interval again in line 10. This is because a previous page reference related to current page

```
ALGORITHM: Cut&Pick($L_{IP}$) ——— $L_{IP}$ is an IP sequence of a proxy log;
01) $UT_{active} \leftarrow$ empty set;
02) $P \leftarrow$ the header of $L_{IP}$; Remove $P$ from $L_{IP}$;
03) WHILE ($L_{IP}$ is not empty) {
04)     $R \leftarrow$ the current page reference in $L_{IP}$; Remove $P$ from $L_{IP}$;
05)     IF ($R.time - P.time > t_{out}$) {
06)         WHILE ( $\exists$ $UT \in UT_{active}$ and $UT.length > 1$) Output $UT$;
07)         $UT_{active} \leftarrow$ empty set;
08)     }ELSE IF ( $\exists$ $UT \in UT_{active}$ and $UT.cluster = R.cluster$){
09)             $P \leftarrow$ the tail of $UT$;
10)             IF ($R.time - P.time <= t_{out}$) Attach $R$ to the tail of $UT$;
CONTINUE;
11)             }
12)     Start a new user transaction $UT_{new} \leftarrow \{R\}$;
13)     Add $UT_{new}$ to $UT_{active}$;
14)     $P \leftarrow R$;
15) }
16) WHILE ( $\exists$ $UT \in UT_{active}$ and $UT.length > 1$) Output $UT$;
```

Fig. 4. Cut-and-Pick

reference may actually be far apart, even when the time interval between current two consecutive pages references in the IP sequence satisfies the interval criteria. On the contrary, when the current two consecutive references do not satisfy the interval criteria, no subsequent reference will satisfies the interval criteria with previous references that has already been checked. Thus our algorithm deactivates all the current active transaction for output (line 07). Noise removal is done on output by checking every transaction's length. If a transaction contains only one page reference, it is regarded as noise and can thus be excluded from the final transaction set.

6 Experiments

Our method for evaluating the effectiveness of our approach in transaction identification is through trace-driven simulation using a set of real-world proxy trace data from Digital Equipment Corporation [Coo96]. We first apply the approach presented by Yang et al. [YZL01] to establish an association-based prediction models from the user transaction set that we identified. Then, based on the prediction models, we experimentally assess the prediction accuracy. The evaluation compares the prediction-model accuracy using transactions derived by time-outs only approach, server-based approach and cut-and-pick approach.

We first perform some pre-processing over the traces. In particular, we removed those requests that are obviously generated by programs running on a

Table 1. Data Sets

Data Set	♯Requests	♯IPs	♯Servers	Size(KB)	Testing Size
1	110,019	302	4,295	4,960	85,963
2	231,116	1,003	7,838	10,356	192,909
3	162,204	2,425	1,003	7,303	256,083
4	379,224	3,517	11,317	16,866	740,832

client. In addition, we do not include those requests are queries with “?” in the URLs and “CGI-BIN” requests. Table 1 gives an overview of four data sets for presenting our results. Specifically, Data set 1 and Data set 2 contain sequences of references by relatively small number of clients (w.r.t different IP addresses) against a relatively lager Web site community. Data Set 3 is a sequence of references where the number of Web sites is relatively small. Data Set 4 is full day logs. In all experiments, we use one day's data set for training the prediction model, and use the next day's logs for prediction.

6.1 Experimental Results

Three transaction identification approaches are examined in this experiment. One is the simplistic approaches using a fixed time interval, denoted by “client” in the figures. For this experiment, the associated time interval threshold is set to 30 minutes for all of the four data sets. The other two are server-based approach and cut-and-pick approach. “server” denotes the approach that each single Web server picks transactions from IP sequences. “cluster” denotes the approach that exploits Web site cluster information. In both of these two approaches, the associated time interval threshold is also 30 mins to be consistent. We compare these three approaches by constructing corresponding prediction models from the transaction sets that they identified. For presentation convenience, we refer to the three models as the *client model*, the *server model* and the *cluster model* respectively in the follow context.

In the first experiment, we quantitatively examined the effect of the three approaches on transaction identification, through constructing association rules from each transaction sets identified. We conduct experiments on all of the four data sets with various settings of maximum rule length (Figure 5(a)), minimum support (Figure 5(b)) and minimum confidence (Figure 5(c,d)). We found that, for all settings, the *cluster model* allow a larger number of rules to be discovered by our association rule mining algorithm. All the four data sets exhibit consistent results in increasing the rule-size.

In addition, we made three observations. First, *Cut-and-pick* approach is effective in identifying long transactions. As we can see in Figure 5(a), when the

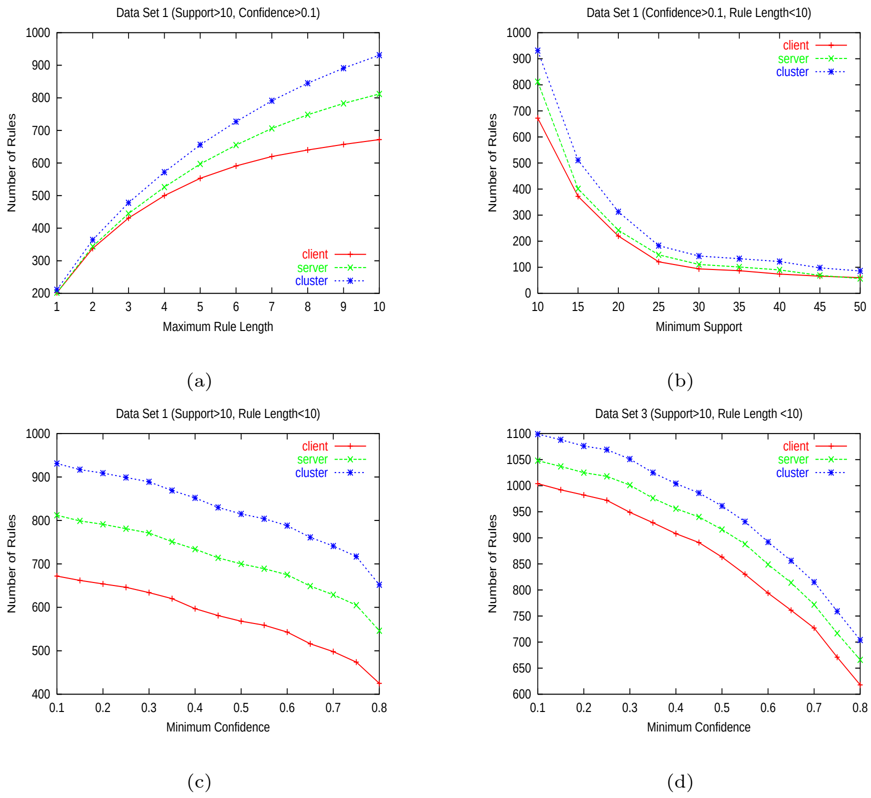


Fig. 5. Number of Rules w.r.t Rule Length/Support/Confidence

rule length increases, the increase in the number of rules become more obvious. This means that there are more longer rules than without using the cut-and-pick method. Second, in Figure 5(a), when the minimum support increases, the curve gets flatter. This is because in proxy logs, there are many low support rules which are typically more vulnerable to noise in the sequence. Third, the mean confidence of all effecting rules in predicting future accesses is also slightly higher in the cut-and-pick approach than in the other two approaches, as illustrated in Figure 8(a). This suggests that exploiting Web site information can be beneficial in extracting appropriate user transactions. Figure 5 also showed that *Cut-and-Pick(server)* can discover more rules than the simplistic methods, suggesting that considerable improvement in association rule discovery can be achieved by separating page references on different sites in the process of identifying user transactions.

The second experiment examines the effectiveness of resulting prediction model in terms of number of hits and prediction accuracy. First, we consider

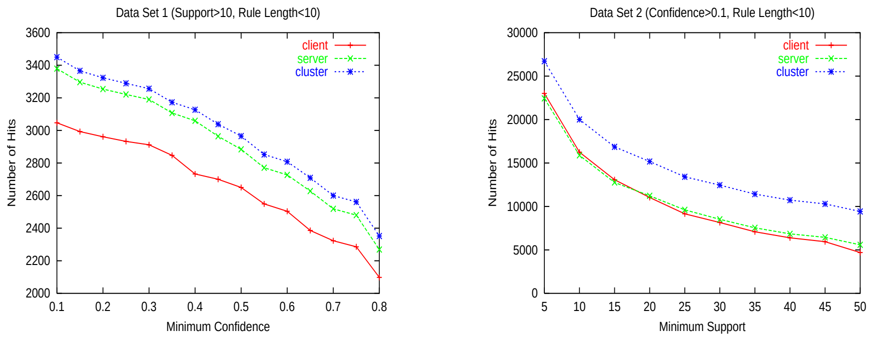


Fig. 6. Number of Hits w.r.t Minimum Confidence/Support

there is a **match** when the current references sequence matches the left hand side of any rule in the model. When there are multiple matches found for one sequence, principle of *longest-match* will be applied [LYW01]. Furthermore, if the next reference in the sequence also matches the right hand side of that particular rule, we count it as a **hit**, i.e., a correct prediction. **Accuracy** of prediction is calculated by dividing the total number of hits by the total number of matches. While the number of hits is used as a quantitative measurement, accuracy is used for qualitatively assessment of a prediction model.

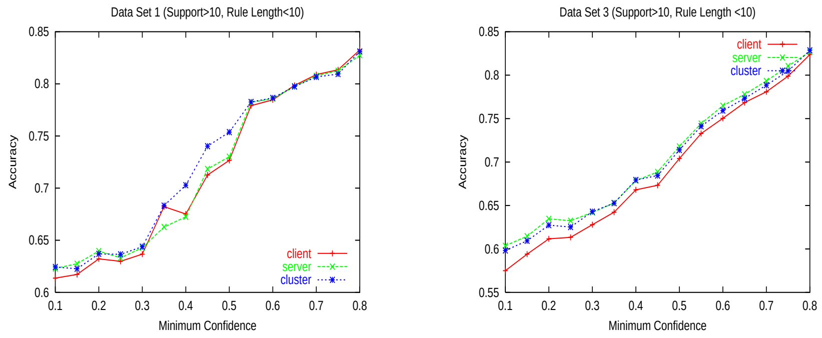
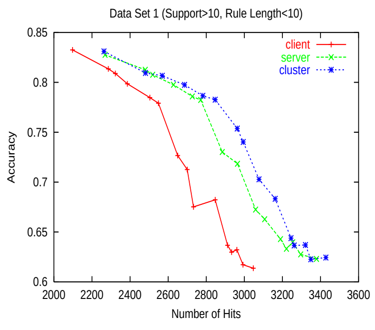
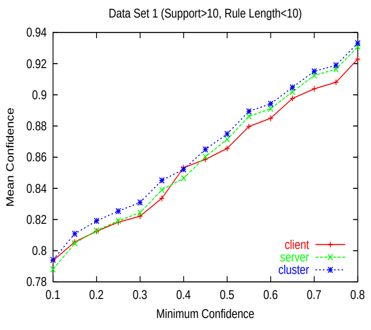


Fig. 7. Accuracy w.r.t Minimum Confidence/Support

As it can be noticed from Figure 6, significant improvement in the total number of hits was found when *cut-and-pick* approaches are used. With minimum confidence varying from 0.1 to 0.8, the number of hits increased by an average of 11.3% for data set 1. Similarly, we also observe increasing improvement up to 35.2% in the number of hits with minimum support ranging from 5 to 50.



(a)



(b)

Fig. 8. (a). Accuracy w.r.t Number of Hits; (b). Mean Confidence of Effecting Rules in Prediction

With increased number of hits, the accuracy of our predictions remains similar to that of the simplistic method. Slight improvement of about 2-5 percentage in accuracy was also found when the confidence is between 0.3 and 0.5 for data set 1, and for the whole range of data set 2. Figure 7 illustrates the accuracy results. In Figure 8(a), we exhibit the plot of both accuracy and number of hits with respect to minimum confidence raging from 0.1 to 0.8. The curve of $Cut\&Pick(cluster)$ is clearly located above other two curves, indicating that $Cut\&Pick(cluster)$ lead to a prediction model that is not only quantitatively better but also qualitatively more accurate.

7 Conclusions

In this paper, we investigate the problem of user transaction identification in proxy logs. In a proxy logs, a single user transaction may include pages references from one site as well as from multiple sites. Moreover, different types of transactions are not clearly bounded and are sometimes interleaved with each other as well as with noise. Thus an effective transaction identifier has to identify interleaved transactions and transactions with noise, and capture both the intra-site transactions and the inter-site transactions. We presented a *cut-and-pick* method for extracting all these transactions, by *cutting* on more reasonable transaction boundaries and by *picking* the right page sequences in each transaction. Our method takes advantages of user behavior that in most transactions, the same user typically visits multiple, related Web sites that form clusters.

We experimentally examine the effectiveness of our approach by using the real-world proxy log data. The results show that our *cut-and-pick* approach can produce more accurate transactions that result in higher accuracy of Web-access prediction models, indicating that relationships among Web sites provide critical information for transaction identification. In the future, we would like to incorporate our approach into proxy caching and prefetching algorithms to improve their performance.

References

ADW01. C. Anderson, P. Domingos, and D. Weld. Personalizing web sites for mobile users. In *Proceedings of the 10th World Wide Web Conference (WWW10)*, Hong Kong, China, May 2-4 2001.

AMS$^+$96. Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining.*, pages 307–328. AAAI/MIT Press, 1996.

AS95. Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proc. 11th Int. Conf. Data Engineering, ICDE*, pages 3–14, Taipei, Taiwan, March 6–10 1995. IEEE Press.

BL99. Jose Borges and Mark Levene. Data mining of user navigation patterns. In *Proc. of the Web Usage Analysis and User Profiling Workshop*, pages 31–36, San Diego, California, 1999.

BL00. J. Borges and M. Levene. A heuristic to capture longer user web navigation patterns. In *Proc. of the first International Conference on Electronic Commerce and Web Technologies*, Greenwich, U.K., September 2000.

CMS99. Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

Coo96. Digitial Equipment Cooperation. Digital's web proxy traces. Available at ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html, 1996.

CPY96. M.-S. Chen, J. S. Park, and P. S. Yu. Data mining for path traversal patterns in a web environment. In *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS)*, pages 385–393, Hong Kong, May 27-30 1996.

CSM97. R. Cooley, J. Srivastava, and B. Mobasher. Web mining: Information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Newport Beach, CA, November 1997.

DH99. J. Dean and M. Henzinger. Finding related pages in the world wide web. In *Proceedings of the 8th International World Wide Web Conference*, pages 1467–1479, Toronto, Canada, 1999.

LYW01. Tianyi Li, Qiang Yang, and Ke Wang. Classification pruning for web-request prediction. In *Proceedings of the 10th World Wide Web Conference (WWW10)*, Hong Kong, China, May 2-4 2001.

NM00. Alexandros Nanopoulos and Yannis Manolopoulos. Finding generalized path patterns for web log data mining. In *Proceedings of East-European Conference on Advances in Databases and Information Systems*, pages 215–228, 2000.

PE00. Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1–2):245–275, 2000.

PM96. Venkata N. Padmanabhan and Jeffrey C. Mogul. Using predictive prefetching to improve World-Wide Web latency. In *Proceedings of the SIGCOMM '96 conference*, 1996.

PM98. T. Palpanas and A. Mendelzon. Web prefetching using partial match prediction. Technical Report CSRG-376, Departement of Computer Science, University of Toronto, 1998.

PP99. James E. Pitkow and Peter Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *USENIX Symposium on Internet Technologies and Systems*, 1999.

PPR96. Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: Extracting usable structures from the web. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI.* ACM Press, 1996.

SKS98. S. Schechter, M. Krishnan, and M. Smith. Using path profiles to predict HTTP request. In *Proceedings of 7th International World Wide Web Conference*, Brisbane, Australia, April 14-18 1998.

SP01. M. Spiliopoulou and C. Pohle. Data mining for measuring and improving the success of web sites. *Data Mining and Knowledge Discovery*, 5(1/2), 2001.

SPF99. Myra Spiliopoulou, Carsten Pohle, and Lukas Faulstich. Improving the effectiveness of a web site with web usage mining. In *Proc. of the Web Usage Analysis and User Profiling Workshop*, pages 51–56, San Diego, California, 1999.

SYZ00. Z. Su, Q. Yang, and H. Zhang. A prediction system for multimedia prefetching on the internet. In *Proceedings of the ACM Multimedia Conference*, October 2000.

YZL01. Qiang Yang, Haining Henry Zhang, and Tianyi Li. Mining web logs for prediction models in www caching and prefetching. In *Proc. of the 7th ACM SIGKDD'01*, San Francisco, California, USA, August 2001.