Web-log Mining for Quantitative Temporal-Event Prediction

Qiang Yang[1], Hui Wang[1] and Wei Zhang[2]

***Abstract*— The web log data embed much of web users' browsing behavior. From the web logs, one can discover patterns that predict the users' future requests based on their current behavior. These web data are very complex due to their large size and sequential nature. In the past, researchers have proposed different methods to predict what pages will be visited next based on their present visit patterns. In this paper, we extend this work to discover patterns that can predict when these web page accesses will occur. Our method is based on a novel extension of association rule classification method. We extend the traditional association rules by including the temporal information explicitly in each rule, and reason about the confidence of each prediction in terms of its temporal region. We compare two different methods for temporal event prediction, demonstrate the effectiveness of our methods empirically on realistic web logs, and explore the tradeoff between prediction accuracy and data mining time for our models.**

***Index Terms*— Web Log Mining, Quantitative Predictions for Web Accesses.**

I. Introduction

THE rapid expansion of the World Wide Web has created an unprecedented opportunity to disseminate and gather information online. As more data are becoming available, there is much need to study web-user behaviors to better serve the users and increase the value of enterprises. One important data source for this study is the web-log data that traces the user's web browsing. In this paper, we study prediction models that predict the user's next requests as well as when the requests are likely to happen, based on the web-log data. The result of accurate prediction can be used for recommending products to the customer, suggesting useful links, presending, pre-fetching and caching web pages for reducing access latency [11], [18].

An important class of data mining problems is mining sequential association rules from web log data. The web-log data consists of sequences of URLs requested by different clients bearing different IP addresses. Association rules can be used to decide the next likely web page requests based on significant statistical correlations. In the past, sequential association rules [3], [2] have been used to capture the co-occurrence of buying different items in a supermarket shopping. Episodes were designed to capture significant patterns from sequences of events [8]. However, these models were not designed for the prediction task, because they do not specify how to select among multiple predictions for a given observation. The works by [6], [17] considered using association rules for prediction by selecting rules based on confidence measures, but they did not consider the classifiers for sequential data. In the network system area, n-gram or path based rules have been proposed for capturing long paths that occur frequently [12], [16], but the researchers in these areas did not study the models in the context of association rules, and offered no comparison with other potential prediction models in a systematic way. As a result, it remains an open question how to construct association rules that predict not only what is likely to happen next given the current observed events, but when these events will occur.

In this paper, we present a quantitative model for temporal event prediction on the web. By quantitative we mean the ability to predict a time interval in which the next web page visits will occur. For example, after observing a web user visiting pages A and B in a row, our system might predict that page C is most likely to be the next page to be visited by the user, and that C is most likely to be visited within 10 to 20 seconds from the current time. Our approach is to extend the traditional association rules by including additional constraints and representations. In web-access prediction, previously used representations often state that if access to pages A, B and C are observed, then D will be predicted to occur next. We extend this representation by including, on the right hand side of each rule, a probable temporal region [t1, t2] in which D will happen, and a confidence estimate on when D will occur. In addition, we place the restriction that the left-hand-side of the rule A, B and C must occur next to each other and in that order; in essence, A, B, C is a substring. The right hand side to be predicted corresponds to a web page access that occurs frequently enough and falls into the specified window with high probability.

The main contribution of the work is the time-accuracy tradeoff between two different methods for web log mining. The first method is based on a minimal-temporal-region heuristic. This method has been studied in AI in the past [19], [20], where an effective solution has been proposed for assembly-line event sequences. Here we generalize this method to web logs, taking into account the special properties in the web logs. Our generalization allows the left-hand-sides of rules to be greater than one, which enables more accurate predictions. Our second method is aimed at achieving efficiency while sacrificing accuracy slightly; it is based on the computation of confidence intervals of normal distributions

1. Department of Computer Science, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong, China,(qyang,whui)@cs.ust.hk

2. The Boeing Company, Seattle, WA USA, Wei.zhang@boeing.com

in temporal events. The minimal-temporal-region method is shown to be more accurate but takes longer in the learning phase. On the other hand, the confidence-interval-based method is more efficient to mine, but is less accurate. Our study offers the web-log mining system designers a choice in algorithms according to the needs in their application domains.

The paper is organized as follows. Section II discusses rule-representation methods. Section III discusses region representation methods. Section IV presents the experimental results. Section V discusses related work. Section VI concludes the paper with a discussion of future work.

II. Rule Representation and Selection

A. Web Logs and User Sessions

Consider the Web log data from a NASA Web server shown in Table I. Typically, these web server logs contain millions of records, where each record refers to a visit by a user to a certain web page served by a web server. This data set contains one month worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The log was collected from 00:00:00 August 1, 1995 through 23:59:59 August 31, 1995. In this period there were 1,569,898 requests. There are a total of 72,151 unique IP addresses, forming a total of 119,838 sessions. A total of 2,926 unique pages were requested.

TABLE I
Example Web Log

kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:17 -0400] "GET / HTTP/1.0" 200 7280
kgtyk4.kj.yamagata-u.ac.jp - - [01/Aug/1995:00:00:18 -0400] "GET /images/ksclogo-medium.gif HTTP/1.0" 200 5866
d0ucr6.fnal.gov - - [01/Aug/1995:00:00:19 -0400] "GET /history/apollo/apollo-16/ Apollo-16.html HTTP/1.0" 200

Given a web log, the first step is to clean the raw data. We filter out documents that are not requested directly by users. These are image requests or CSS requests in the log that are retrieved automatically after accessing requests to a document page containing links to these files and some half-baked requests. Their existence will not help us to do the comparison among all the different methods.

We consider web log data as a sequence of distinct web pages, where subsequences, such as user sessions can be observed by unusually long gaps between consecutive requests. For example, assume that the web log consists of the following user visit sequence: (A (by user 1), B (by user 2), C (by user 2), D (by user 3), E (by user 1)) (we use "()" to denote a sequence of web accesses in this paper). This sequence can be divided into user sessions according to IP address: Session 1 (by user 1): (A, E); Session 2 (by user 2): (B, C); Session 3 (by user 3): (D), where each user session corresponds to a user IP address.

In deciding on the boundary of the sessions, we studied the time interval distribution of successive accesses by all users, and used a heuristic splitting method for a new session. We will present this method in detail in Section V.

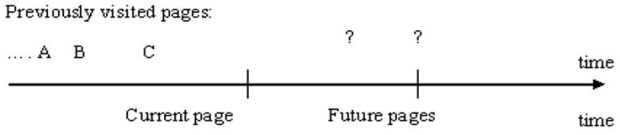


Fig. 1. Moving Window Illustration

To capture the sequential and time-limited nature of prediction, we define two windows. The first one is called *antecedent window*, which holds all visited pages within a given number of user requests and up to a current instant in time. A second window, called the *consequent window*, holds all future visited pages within a number of user requests from the current time instant. In subsequent discussions, we will refer to the antecedent window as W_1, and the consequent window as W_2. Intuitively, a certain pattern of web pages already occurring in an antecedent window could be used to determine which documents are going to occur in the consequent window. Fig 1 shows an example of a moving window.

The moving windows define a table in which data mining can occur. Each row of the table corresponds to the URL's captured by each pair of moving windows. The number of columns in the table corresponds to the sizes of the moving windows. Table II shows an example of such a table corresponding to the sequence (A, B, C, A, C, D, G), where the size of W_1 is three and the size of W_2 is one. In this table, under W_1, A1, A2 and A3 denote the locations of the last three objects requested in the antecedent window, and "Prediction" and "Time Interval" are the objects and predicted time interval in the consequent window.

TABLE II
A portion of the Log Table extracted by a moving window pair of size [3, 1]

W1			W2	
A1	**A2**	**A3**	**Prediction**	**Time Interval**
A	B	C	A	[10 min, 20 min]
B	C	A	C	[5 min, 15 min]
C	A	C	D	[3 min, 4 min]

B. Prediction Rule Representation

We now discuss how to extract rules of the form $LHS \rightarrow RHS$ from the session table. Our different methods will extract rules based on different criteria for selecting the LHS. However, we restrict the RHS in the following way. Let U1, U2, ... Un be the candidate URL's for the RHS that can be predicted based on the same LHS. We build a rule $LHS \rightarrow \langle Uk, [t1, t2] \rangle (supp, conf)$ where the URL Uk occurs most frequently in the rows of the table among all Ui's

in the set U1, U2, ... Un. [t1, t2] is the region determined by the data mining algorithm in which the event Uk is most likely to occur, and *supp* and *conf* are the support and confidence for such occurrence, respectively, in Equations 2 and 2 below.

The rule representation we use is known as the *latest-substring rules*. These rules not only take into account the order and adjacency information, but also the *recency* information about the LHS string. In this representation, only the substrings ending in the current time (which corresponds to the end of the window W_1) qualifies to be the LHS of a rule. These are also known as hybrid n-gram rules in some literature [12], [16]. For example, Table III shows the latest-substring rules example.

TABLE III
LATEST-SUBSTRING RULES. T1, T2 AND T3 ARE TIME INTERVALS

W1	W2	Latest substring Rules
A, B, C	D	⟨A, B, C⟩ → ⟨D, T1⟩, ⟨B, C⟩ → ⟨D, T2⟩, ⟨C¿ → ⟨D, T3⟩

Viewed from another angle, latest-substring rules could also be considered as the union of N^{th}-order Markov models [9], where N covers different orders up to the length of W_1. Therefore, it is more general than the N-gram models or N^{th}-order Markov models. However, through our other experiments, we have found out that the Markov models' performance drops when N exceeds a certain threshold, but the latest-substring method that considers multiple Nth-order models for different N demonstrates a monotonically increasing precision curve.

For any given set of rules, we also have an option to add a default rule that captures all cases where no rule in the rule set applies; when no LHS of all rules apply to a given observed sequence of URL's, the default rule always applies. For example, a default rule can simply be the most frequently requested page in the training web log.

For each rule of the form LHS→ RHS, we define the *support* and *confidence* as follows

$$labeleq2sup = \frac{count(LHS, RHS)}{count(Table)} \quad (1)$$

$$conf = \frac{\sup(LHS, RHS)}{\sup(LHS)} \quad (2)$$

In the equations above, the function *count*(*Table*) returns the number of rows in the log table, and *count*(*LHS*) returns the number of rows in the log table that W_1 is a certain LHS.

$$sup(LHS) = \frac{count(LHS)}{count(Table)} \quad (3)$$

C. Rule-Selection Methods

In classification, Liu et a. [6] extended association rules to build confidence-based classifiers. In this section, we extend their work further by including rankings on temporal intervals. Our goal is to output the best guess on a class based on a given observation. In different rule-representation methods, each observation (or case) where the LHS matches the case can give rise to more than one rule. Therefore, we need a way to select among all rules that apply. In a certain way, the rule-selection method compresses the rule set; if a rule is never applied, then it is removed from the rule set. The end result is that we will have a smaller rule set with higher quality. In addition to the extracted rules, we also define a default rule, whose RHS is the most popular page in the training web log and the LHS is the empty set. When no other rules apply, the default rule is automatically applied.

For a given set of rules and a given rule-selection method, the above rule set defines a classifier. With the classifier, we can make a prediction for any given case. For a test case that consists of a sequence of web page visits, the prediction for the next page visit is correct if the RHS of the selected rule occurs in window W_2. For N different test cases, let C be the number of correct predictions. Then the precision of the classifier is defined as

$$precision = \frac{C}{N} \quad (4)$$

Our rule selection method is called the most-confident selection method. It always chooses a rule with the highest confidence among all the applicable association rules. A tie is broken by choosing a rule with a longer LHS. For example, suppose that for a testing case and antecedent window of size four, an observed sequence is (A, B, C, D). Suppose that using the most-confident rule selection method, we can find three rules which can be applied to this example, including:

Rule 1: (A, B, C, D) → E, T1 with confidence 30%
Rule 2: (C, D) → F, T2 with confidence 60%
Rule 3: (D) → G, T3 with confidence 50%.

In this case, the confidence values of rule 1, rule 2 and rule 3 are 30%, 60% and 50%, respectively. Since Rule 2 has the highest confidence, the most-confident selection method will choose Rule 2, and predict F.

The rationale of most-confident selection is that the testing data will share the same characteristics as the training data that we built our classifier on. Thus, if a rule has higher confidence in the training data, then this rule will also show a higher precision in the testing data. As we will see, this assumption is not always correct, as it can lead to overfitting rules. However, this problem can be solved by introducing a filtering step, which removes all rules for which the support value is below a threshold. In our experiment, we used a support threshold of value 10.

Note that a rule may have different regions with different confidence values. Each region is associated with a different confidence value. In addition, we allow the RHS of a rule to predict more than one URL, in the decreasing value of

the confidence strength. For example, we might have a rule whose LHS is "A, B, C" and whose RHS is {(D, [t1, t2], conf1, supp1), (F, [t3, t4], conf2, supp2)}. In this case, our prediction algorithm can predict up to n events that might occur in the future, including D, F, etc. This is known as the n-best method. For this method, if one of the predicted URL occurs in the corresponding range, then a hit is registered towards final precision calculation.

We performed several experiments to show the effects of n-best prediction. Figures 2 and 3 show the trend as n increases. The three dotted lines correspond to three region-selection methods, which we will explain in detail in Section III. It is clear from the figures that, when we set n to be two, the precision is already high enough. When n is greater than two, there is not much improvement. This result tells us that typically each association rule needs only consider the top two best predictions among all possible predictions.

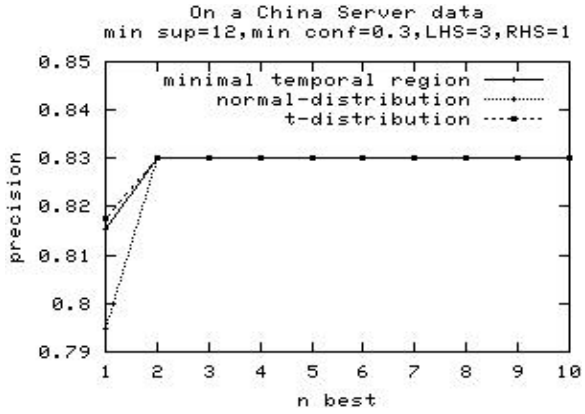


Fig. 2. Precision as the n increases on a China Web data

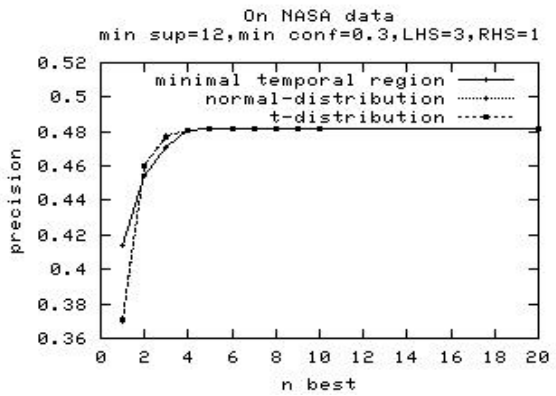


Fig. 3. Precision as the n increases on NASA data

III. Temporal Region Representation Methods

Now, we will describe how to choose a certain temporal region for rule construction. We have two families of region selection methods: a confidence interval based method and a minimal region selection method.

A. Confidence Interval Method

Consider a prediction: when (A, B, C) occurs, what is the next event that is likely to occur? Furthermore, if we decide that D is most likely to occur next, when will D occur? We are interested in computing a time interval $[t_1, t_2]$ meaning that D is likely to occur in the future between t_1 and t_2 time scope. We also would like to place a high level of confidence on this interval prediction; for example, we might choose a 95% confidence. In order to make the prediction, we will collect all the association rules of the form: LHS$\rightarrow \langle$RHS, $[t_1, t_2]\rangle$ (supp, conf) from the training data. The task in this section is how to get $[t_1, t_2]$ that is both accurate and narrow to be useful.

Our method is to compute the set of time lags in which D occurs after A, B and C occurs. This collection of time points is called the lag-set. For an example, a rule:

(A, B, C)$\rightarrow \langle$D, $[t_1, t_2]\rangle$ (supp, conf)

has a lag-set {4, 9, 20, 22, 31, 39, 39, 39, 40, 41, 41, 42, 43, 45, 53, 61}

This means that when A, B, C are observed to occur next to each other, D occurred at 4 seconds after C, 9 seconds after C, and 20, 22, 31, 39 seconds after the occurrence of C. Note that we have three '39"s here, denoting that 'D' occurred three times at 39 seconds after C. *Supp* and *conf* are this rule's support and confidence information.

A naïve time interval for this rule is to choose $[t_1, t_2]$ to be: [4, 61], corresponding to the first and last time points of the lag set. However, we could do much better. From the lag-set such as the one above, we can draw an occurrence density curve. If the lag-set is large enough, we may expect the curve to demonstrate the standard normal distribution as shown in Fig 4. Thus, we can use the normal distribution formulas to choose an interval $[t_1, t_2]$.

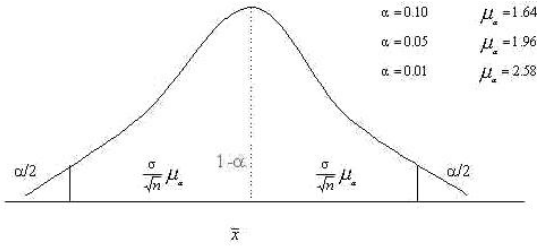


Fig. 4. Standard Normal Distribution

In classical statistics theory, for a large data set, we use formula (5) to measure the confidence interval.

$$[\bar{x} - \frac{\sigma}{\sqrt{n}}\mu_\partial, \bar{x} + \frac{\sigma}{\sqrt{n}}\mu_\partial] \quad (5)$$

where $\bar{x}$ is the mean, σ^2 is the standard deviation, μ_∂ follows the normal distribution table. n is the number of examples in the training data that supports this interval.

For the example above, if n=16, $\bar{x} = 35.56$, $\sigma = 15.6$, then for the confidence level 1-0.05=95% ($\partial = 0.05$), $\mu_\partial = 1.96$. Hence, the temporal region will be [28.21,42.91].

For a small data set, we use the t-distribution formulas instead, and changing σ^2 to s^2(s^2 is variance deviation), and μ_∂ to $t_\partial(n-1)$; the latter $(n-1)$ is a number in the t-distribution table equal to $t_\partial(n-1)$. The interval is chosen according to formula (6):

$$[\bar{x} - \frac{s}{\sqrt{n}}t_\partial, \bar{x} + \frac{s}{\sqrt{n}}t_\partial] \quad (6)$$

As an example, let the confidence level be 1-0.05 = 95%, $t_\partial(n-1) = t_\partial(15)$ = 1.753. Thus the region is [28.96, 42.16].

B. Minimal Temporal Region Selection

The confidence-interval-based method presented above chooses an interval based on the confidence region. However, it does not express our wish to find a temporal region that is as narrow as possible while covering as many training cases as possible. A minimal temporal region method was proposed in [19] and [20]. However, in these works it is required that each rule's LHS has a size of one. In this section, we extend this method to include association rules whose LHS can be greater than one.

A minimal temporal region is the smallest time interval that covers all the values in a subset of a lag set. Consider an example, suppose that a rule: (A, B, C) $\rightarrow$ D has a lag set {0,17,62,87). This will result in 10 temporal regions: [0,0], [0,17], [0,62], [0,87], [17,17], [17,62], [17,87], [62,87] and [87,87]. Our aim is to choose a temporal region from the above with the smallest scope and covers all occurrences. We use a heuristic in formula (7) to obtain a score for each of these regions:

$$Score = W_1 * Accp + W_2 * Rng + W_3 * Cov \quad (7)$$

Intuitively, this formula is trying to balance three factors: high accuracy, short range of the time interval and large coverage. The region with the highest score will be chosen. The definitions for Accp, Rng, Cov are as follows:

1. Prediction Accuracy (Accp): this factor computes the percentage of cases that a target event occurs in the time region over all cases that a condition event occurs.

2. Range (Rng): While Accp reward large regions (their values increase monotonously as the size of a temporal region grows), Rng is a factor encouraging smaller regions, is defined as 1-Intv(r)/(MaxLag-MinLag+1), where Intv(r) is the region size of rule r.

3.Coverage (Cov): This computes the rate of cases covered by a rule over all cases that are covered by the same condition-target pair but with the full search scope defined by MinLag and MaxLag. We denote the latter as AllCntScp. Then the Cov is AllCnt/AllCntScp.

The weights W_1, W_2 and W_3 express their relative importance. In general, they can be learned using linear regression method. We set them to one in our experiments.

IV. Experimental Results

In the last section, we presented two methods for temporal region computation. The first method is a confidence-based method, based on the assumption that the event distribution follows a normal distribution. The advantage of this method is that it requires one scan of the time points in the web log, resulting in linear time complexity in computation. However, the normal-distribution assumption is quite a strong one. The minimal temporal region heuristic, on the other hand, does not make this assumption. Instead, it looks for a good trade off point between the coverage, size and accuracy of the time points in the web log. The price to pay is that it involves more computation.

In this section, we will explore the relative merits of these two methods in detail. The tradeoff between accuracy and computation time studied here corresponds to the main contribution of this paper.

A. Experimental Setup

Our goal is to select a best rule representation with the region. We employ 3 realistic data sets: NASA, EPA and a new data of Web Server located at China. EPA log was collected from 23:53:25 on Tuesday, August 29 1995 to 23:53:07, August 30 1995, about 4.8M. After removing some irregular logs, we have 2225 unique visiting IP address, and 4149 unique pages are requested and 17933 requests. The NASA data is described in Section 2. We also used a more recent dataset from a penpal-service portal site located in Beijing. It was collected from 00:00:00 Jan 22, 2002 to 21:12:44 Jan 22, 2002, with a size of about 7.8M. After data cleaning, we have 270 unique IP address, 1000 unique web pages and 9688 requests. In this experiment, requests on the same CGI with different parameters are considered as different pages. For example: "/htbin/wais.pl?STS-59" and "/htbin/wais.pl?IMAX" are two pages in our system.

To obtain user sessions, we use a heuristic user-session splitting method. The heuristic is to calculate the mean of the gaps between two consecutive requests in the web logs. For each next page request, if the time gap is larger than a constant number of the ancestral mean time gap , we consider the request as starting a new session. For example, we use 70 as the constant factor in subsequent experiments.

In our experiment, we split all the sessions into training set and testing set by splitting the data into two equal parts and then construct the association rules from the training data. We restrict the LHS sizes to be no larger than three and RHS size to be one. Our filtering method removed all rules for which the confidence is less than a minimal confidence and the support is less than a minimal support. The minimum confidence and support values are used as variables in our tests in the following sections to test their effectiveness.

B. Comparison on Precision

Table IV provides a comparison of precision of the naive temporal-region selection based methods. When a default rule is used, where the default rule is defined as the most popular page in the web log, the default prediction is made whenever no rule whose LHS matches the current observation in the test data.

The 'Precision with default rule' is defined as:

$$Precision = \frac{C_{all}}{N_{test\_cases}} \quad (8)$$

The set of N_{test_cases} test cases is a section of the web log where each test case corresponds to a user access to a web page. C_{all} corresponds to the set of all correct guesses according to our prediction; this set is also known as all the correct "hits".

The 'Precision without default rule' is defined as:

$$Precision = \frac{C_{without\_default\_rule}}{N_{predicted\_cases}} \quad (9)$$

TABLE IV

PRECISION AS n IN N-BEST INCREASES FOR NASA DATA

Confidence-interval Based	n-best	Precision with default rule	Precision without default rule
	n=1	0.35411	0.380741
	n=2	0.42339	0.457589
Minimal temporal region selection		Precision with default rule	Precision without default rule
	n=1	0.38232	0.41987
	n=2	0.41630	0.45866

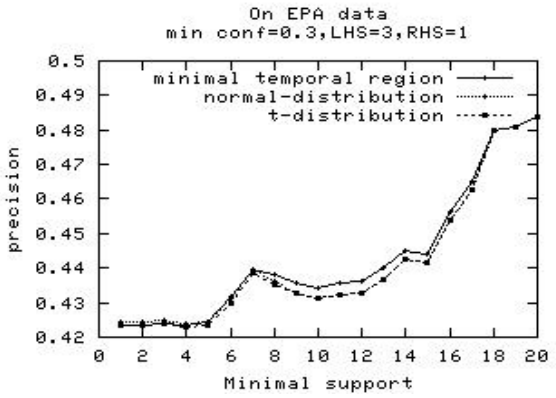
In this test, all data from the NASA log are used, with 50From these results, we conclude that both methods give similar accuracy results, with the minimal temporal-region selection giving slightly better performance when $n = 1$.

C. Comparing Temporal-Region Selection Methods

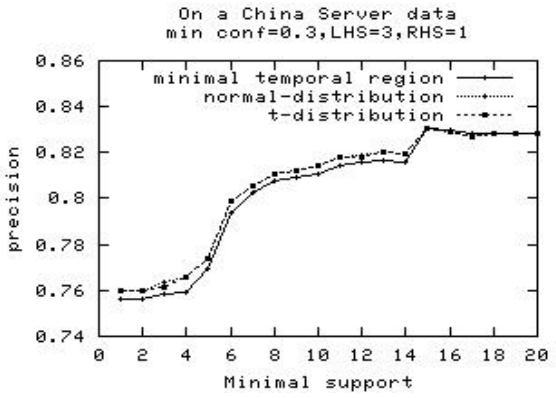
In this section, we will compare the performance of three region-selection methods. For brevity, we only consider prediction without the default rule.

Fig 5a-c show the prediction precision as the minimal support changes. In this test, minimal temporal region is better than the standard normal distribution and t-distribution, especially for large datasets (NASA data). This can be explained by the fact that the standard normal/t distribution methods prefer the largest regions around the mean of each event occurrence. However, the minimal temporal region also prefers small regions by using the factor Rng. Therefore the minimal temporal region method makes a balance between accuracy and narrow time regions. For the China web data, these methods have the similar performance.

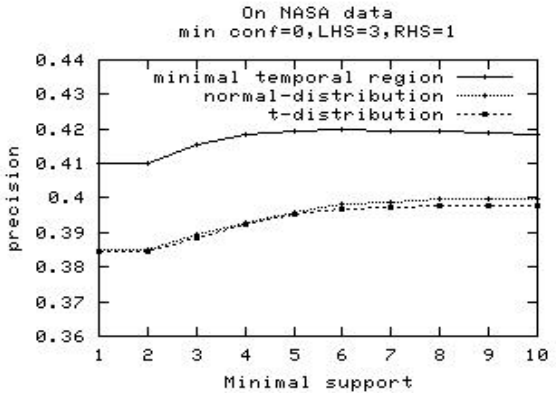
We now consider the performance in terms of accuracy as the minimal confidence increases. The results are shown in Fig 6a-c. The minimal temporal region method is a little better than the other two for NASA data, but is comparable for the other two datasets. The general trend is that the precision will



a. Precision as the Minimal Support Increases on EPA data



b. Precision as the Minimal Support Increases on A China Web data



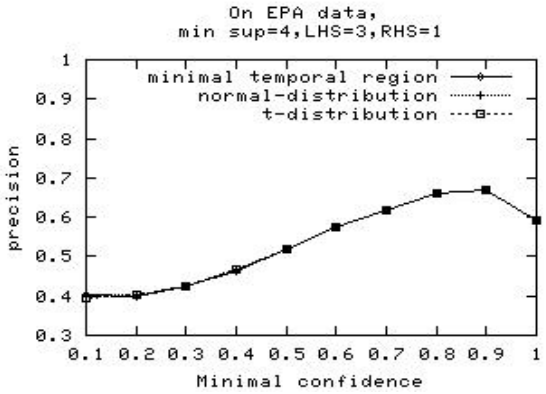
c. Precision as the Minimal Support Increases on NASA data

Fig. 5. Precision results on three data sets

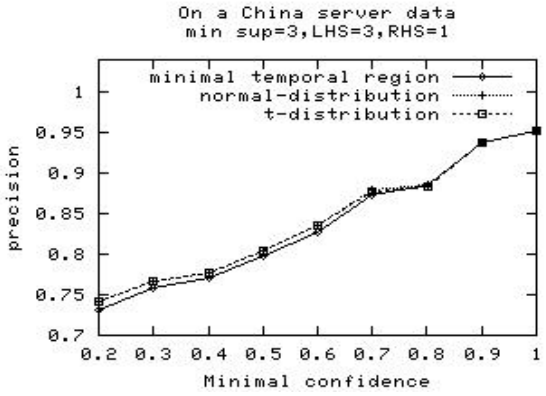
increase as the minimal confidence increases. The decrease in EPA data when minimal confidence is one

We next varied the size of LHS. We set min_conf=0.6, min_sup=15 for the China data and min_conf=0.3, min_sup=12 for the NASA data. The results are in Fig 7a and 7b. As the LHS size increases, the prediction precision first increases, then decreases, especially for China web data. This is because as larger LHS rules are admitted, more overfitting rules are also admitted. These rules typically have high confidence. Thus, there is a decrease in precision when LHS past 3.

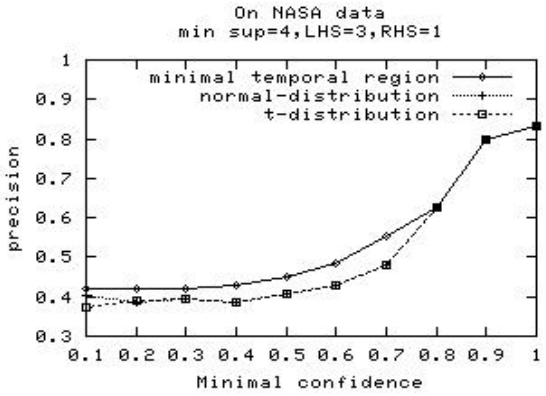
We also tallied the number of rules in our rule sets with different sized LHS's. This allows us to show what proportion of the predictions benefited from rules of different lengths. The results are shown in Table V. The n% is the defined as



a. Precision as the Minimal Confidence increases on EPA data

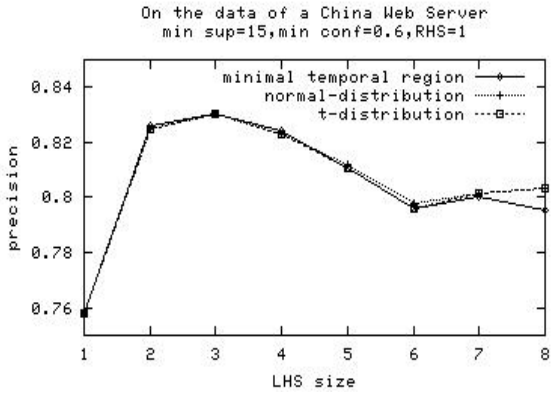


b. Precision as the Minimal Confidence Increases on a China Web data

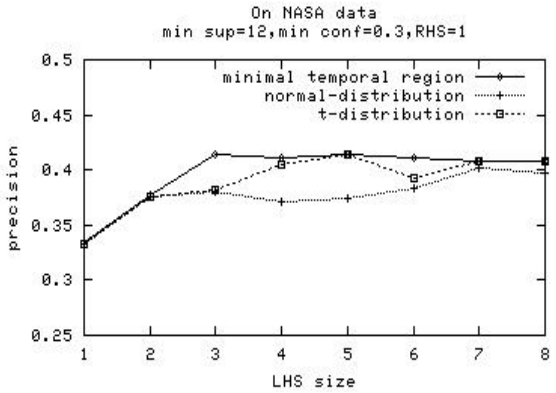


c. Precision as the Minimal Confidence Increases on NASA data

Fig. 6. Precision results on three data sets



a. Precision as the LHS Size Increases on a China web Server data (set min_conf=0.6, min_sup=15)



b. Precision as the LHS Size Increases on NASA data (set min_conf=0.3, min_sup=12)

Fig. 7. Precision results as the LHS Size Increases

the proportion of LHS size over all the predicted instances. From Table V, we can see that for all methods, the majority of rules are still length-one rules. However, there is a significant number of length-two rules as well.

Finally, the time complexity of the confidence-based model in our web-log mining problem is only linear in the length of the web logs. For each antecedent window W_1, the number of consecutive substrings that end with the end of the window is of size $|W_1|$. Thus, the number of LHS's that need be examined is only $O(|W_1| * |WebLog|)$. In contrast, the minimal temporal-region selection method takes $O(|Weblog|^2)$ time to mine in the worst case, since for each event E which occurs N times, it computes all $O(N^2)$ time intervals before selecting a best one according to formula (7).

V. Related Work

Much recent research activity in sequence prediction falls into the research areas of data mining and computer networks. In the data mining area, most algorithms are designed to deal with a database consisting of a collection of transactions (see [13] for example). These records store the transaction data in applications such as market-basket analysis. The focus of research has been how to perform efficient and accurate association and classification calculations.

In data mining area, general classification algorithms [13] were designed to deal with transaction-like data. Such data has a different format from the sequential data, where the concept of an attribute has to be carefully considered. As shown in this paper, these algorithms can be used to build the prediction models by applying a 'moving-window' algorithm across the whole web log sequence, such that the transactions appearing together in the same window can be regarded as a record in transaction data.

Association is another extensively studied topic in data mining. Association rules [3] were proposed to capture the co-occurrence of buying different items in a supermarket shopping. It is natural to use association rule generation to

TABLE V
DISTRIBUTION OF THE LHS SIZES FOR RULES USED IN THE PREDICTION (MIN_SUP=10, MIN_CONF=0.3, N-BEST=1, LHS¡=3, RHS=1)

		LHS=1	LHS=2	LHS=3
A China Server	Naive	78.5%	17.2%	4.3%
	Minimal temporal region	78.5%	17.2%	4.3%
	Standard Normal distribution	65%	22.9%	12.1%
	t-distribution	65%	22.9%	12.1%
NASA	Naive	35.6%	40%	24.4%
	Minimal temporal region	39.5%	39.6%	20.9%
	Standard Normal distribution	36.1%	39.8%	24.1%
	t-distribution	35.6%	39.4%	25.0%

relate pages that are most often referenced together in a single server session [14], [5]. However, correlation discovery is not sufficient to build a prediction model, because they do not consider the sequential nature of knowledge embedded in web logs. In data mining area, [4], [2] proposed sequential association mining algorithms, but these are designed for discovery of frequent sequential transaction itemsets. They cannot be applied directly for sequence prediction without first being converted to classifiers. [6], [17] considered using association rules for prediction and classification, which achieved observable improvement on accuracy of classification models, but they did not consider sequential data either.

In the network area, researchers have used Markov models and N-grams [16], [12], [15] to construct sequential classifiers. Markov models and N^{th}-order Markov models when parameterized by a length of N, are essentially represent the same functional structure as N-grams. Generally speaking, these systems analyze the past access history on the web server, maps the sequential access information in N consecutive cell series called N-grams, and then builds prediction models. [9] proposed several different ways to build N-gram based models, and empirically compared their performance on real-world web log data. [16], [17] performed empirical studies on the tradeoffs between precision and applicability of different N-gram models, showing that longer N-gram models can make more accurate prediction than shorter ones at the expense of lower coverage. [16] proposed an intuitive way to build the model from multiple N-grams and select the best prediction by applying a smoothing or 'cascading' model, which prefers longer n-gram models. [15] proposed a small variant version of the longest match method by defining a threshold to go down a certain sequential path. [12] suggested a way to make predictions based on K^{th}-order Markov models.

Researchers in Machine Learning [19], [20] have studied the temporal region learning to find event patterns represented in the form of temporal orders and time. Heuristic methods are studied to select the best rules to be applied. However, these methods have only been designed to discover rules for which the left-hand-side has size one, and are tested on artificially designed event sequences that are of small scale. In this paper, we extend the representation to include larger sized rules, and test the rule based prediction results on realistic and large-scale data sets.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we studied different association-rule based temporal region prediction methods for web request prediction. We studied three different methods, the naïve method, the confidence interval based methods and the minimal temporal region method for the prediction. Our conclusion is that the confidence interval based methods and the minimal temporal region methods perform similarly, with the latter being a little better in precision. Our method represents a novel extension of the association rule based classification method for large-sized sequential data.

In the future, we plan to explore more on the relationship between temporal region prediction and other types of classification. We will also try to integrate the different methods. We believe that the confidence interval based method can indeed be enhanced by factors such as the range and coverage factors used in the temporal region prediction methods.

REFERENCES

[1] M.Arlitt, R.Friedrich, L.Cherkasova, J.Dilley, and T.Jin. Evaluating Content Management Techniques for Web Proxy Caches. *HP Technical Report*, Palo Alto, Apr. 1999.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo. Fast discovery of association rules. In *Advances in knowledge discovery and data mining.* 307-328, AAAI/MIT Press, 1996

[3] R. Agrawal and R. Srikant. Fast algorithm for mining association rules. *Proceedings of the Twentieth International Conference on Very Large Databases.* 1994. pp 487-499

[4] R. Agrawal and R. Srikant. Minging Sequential Patterns. In *Proc. of Int'l Conference on Data Engineering,* Taipei, Taiwan, 1995

[5] E. Cohen, B. Krishnamurthy and J. Rexford, Efficient algorithms for predicting requests to web servers. In *Proceedings of the IEEE INFOCOM '99 Conference,* 1999.

[6] B. Liu, W. Hsu, and Y. Ma. Integrating Classification and Association Mining. In *Proc. of the Fourth Int'l Conf. on Knowledge Discovery and Data Mining (KDD-98),* New York, 1998, pp. 80-86.

[7] L. Oreiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. Wadsworth, Belmont,CA, 1984

[8] H. Mannila, H. Toivonen, and I. Verkamo. *Discovering frequent episodes in sequences.* In Proceedings of the First Int'l Conference on Knowledge Discovery and Data Mining (KDD'95), Montreal, Canada, August 1995. AAAI Press. pp. 210 – 215.

[9] A. E. Nicholson, I. Zukerman, and D. W. Albrecht. A Decision-theoretic Approach for Pre-sending Information on the WWW. In *PRICAI'98 - Proc. the Fifth Pacific Rim Int'l Conf. on Artificial Intelligence,* Singapore, page 575-586.

[10] J. Pei, J.W. Han, B. Mortazavi-asl and H. Zhu, Mining Access Patterns efficiently from Web Logs, In *Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD 2000*: 396-407. 2000

[11] M. Perkowitz and O. Etzioni. Adaptive web sites: Concept and case study. In *Artificial Intelligence,* volume 118 (1-2), pages 245–275, 2001.

[12] J.Pitkow and P.Pirolli. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. In *Second USENIX Symposium on Internet Technologies and Systems,* Boulder, C0, 1999.

[13] J.R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann, 1993

[14] J. Srivastava, R. Cooley, M. Deshpande and P. Tan, Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations,* 1(2):12–23, 2000.

[15] S. Schechter, M. Krishnan and M. D. Smith, Using path profiles to predict HTTP requests. In *7th International World Wide Web Conference*, pages 457–467, Brisbane, Qld., Australia, April 1998.

[16] Z. Su, Q. Yang and HJ Zhang, A Prediction System for Multimedia Pre-fetching in Internet. In *ACM Multimedia Conference 2000*, 2000

[17] Q. Yang, H. Zhang and I. T. Li. Mining Web Logs for Prediction Models in WWW Caching and Prefetching . In The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'01, August 2001 San Francisco, California, USA.

[18] Qiang Yang and Henry Haining Zhang. Integrating Web Prefetching and Caching Using Prediction Models. In *World Wide Web Journal. Kluwer Academic Publishers.* Vol. 4 No. 4, 2001. Pages 299-321.

[19] W. Zhang, A region-based learning approach to discovering temporal structures in data, *In Proceedings of the International Conference on Machine Learning.* 1999

[20] W. Zhang, Some Improvement on Event-Sequence Temporal Region Methods. In *Proceedings of the European Conference on Machine Learning.* PP 446-458. 2000