Data Mining and Knowledge Discovery, 8, 253–273, 2004

Building Association-Rule Based Sequential Classifiers for Web-Document Prediction

QIANG YANG* qyang@cs.sfu.ca
TIANYI LI tlie@cs.sfu.ca
KE WANG wangk@cs.sfu.ca
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6

Editors: Fayyad, Mannila, Ramakrishnan

Received October 30, 2001; Revised November 26, 2002

Abstract. Web servers keep track of web users' browsing behavior in web logs. From these logs, one can build statistical models that predict the users' next requests based on their current behavior. These data are complex due to their large size and sequential nature. In the past, researchers have proposed different methods for building association-rule based prediction models using the web logs, but there has been no systematic study on the relative merits of these methods. In this paper, we provide a comparative study on different kinds of sequential association rules for web document prediction. We show that the existing approaches can be cast under two important dimensions, namely the type of antecedents of rules and the criterion for selecting prediction rules. From this comparison we propose a best overall method and empirically test the proposed model on real web logs.

Keywords: web log mining, sequential classifiers, presending web documents

1. Introduction

The rapid expansion of the World Wide Web has created an unprecedented opportunity to disseminate and gather information online. There is an increasing need to study web-user behavior to better serve the web users and increase the value of enterprises. One important data source for this study is the web-server log data that traces the user's web browsing actions. In this paper, we study prediction models that predict the user's next requests based on web-log data. The result of accurate prediction can be used for recommending products to the customers, suggesting useful links, as well as pre-sending, pre-fetching and caching of web pages for reducing access latency.

This paper is aimed at studying how to build sequential classifiers from association rules obtained through data mining on large 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 (Agrawal and Srikant, 1995; Agrawal

*Present address: Department of Computer Science, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong. E-mail: qyang@cs.ust.hk et al., 1996) have been used to capture the co-occurrence of buying different items in supermarket shopping domains. Episodes were designed to capture significant patterns from sequences of events (Mannila et al., 1995). However, these models were not designed for the prediction task, because they do not specify how to select among multiple predictions for a given observed. The works by Liu et al. (1998) and Wang et al. (2000) 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, Markov models have been proposed for capturing browsing paths that occur frequently (Pitkow and Pirolli, 1999; Su et al., 2000). However, researchers in this area did not study the prediction models in the context of association rules, and they did not perform any comparison with other potential prediction models in a systematic way. As a result, it remains an open question how to construct the best association rule based prediction models for web log data.

In this paper, we systematically study different methods for building association-rule based prediction models from web-log data using association rules. We aim at constructing the best prediction model. In this work, we examine two important dimensions of building prediction models, namely, the type of antecedents of rules and the criterion for selecting prediction rules. On the first dimension, we consider *five* types of antecedents, namely subset, subsequence, latest-subsequence, substring, and latest-substring. These representations build the left-hand-side of association rules using non-empty subsets of URLs, non-empty sequences of URLs, non-empty sequences of URLs which end in the current time in the antecedent window, non-empty adjacent sequences of URLs, and non-empty adjacent sequences of URLs which end in the current time in the antecedent window, respectively. These five rule-representation methods cover most existing work in both sequential data mining and computer network areas, as indicated in the related work section.

On the second dimension, we consider three criteria for selecting prediction rules, namely, confidence, accuracy, and matching length. The first criterion will select the applicable rule that has the highest confidence, thus, maximizes the confidence of prediction. The accuracy-based criterion will select the applicable rule that has the highest estimated accuracy on new sequences. This criterion takes into account pruning insignificant rules that over fit the training data. The matching length criterion selects the applicable rule of the longest match antecedent (Pitkow and Pirolli, 1999). A prediction model can be constructed by pairing any antecedent type with any rule-selection criterion. We study the performance of all such prediction models on real life web-log data. Our experiments show that the latest-substring method coupled with the pessimistic-confidence based selection (Quinlan, 1993) gives the best result.

The prediction models that we build are based on web logs that correspond to many users' behavior. Therefore, they are used to make prediction for a general user and are not based on the data for a particular client. They are therefore called *user-independent* association rules. As indicated before, the learning algorithm we will explore shall only require that we can identify a sequence of accesses made by the same users, but we do not need to know the true IP addresses or the identification of the users themselves.

This paper is organized as follows. In Section 2, we review the past works in related research. In Section 3, we present the rule-representation dimension. In Section 4, we present the rule-selection dimension. In Section 5, we perform an analysis of the different methods, and suggest the best one. We conclude our work in Section 6.

2. 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 records (see Quinlan, 1993; Breiman et al., 1984 for example). These records store the transaction data in applications such as supermarkets. The focus of research has been how to perform efficient and accurate association and classification calculations.

In data mining area, general classification algorithms (Quinlan, 1993) 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. The association-rule representation is an extensively studied topic in data mining. Association rules (Agrawal and Srikant, 1994) were proposed to capture the co-occurrence of buying different items in a supermarket shopping. It is natural to use association rule generation to relate pages that are most often referenced together in a single server session (Srivastava et al., 2000). In the data mining area (Agrawal and Srikant, 1995; Agrawal et al., 1996) proposed sequential association mining algorithms, but these are designed for the discovery of frequent sequential transaction itemsets. They cannot be applied directly for sequential prediction problems because they have to be converted to classifiers first; that is, for any given observation we must have a way of deciding which of a collection of application patterns to apply in order to predict what will happen next. This central question is not addressed by the aforementioned work. In contrast, the work of Liu et al. (1998) and Wang et al. (2000) considered using association rules for prediction, but they did not consider sequential data.

In the network area, researchers have been using Markov models and N-grams to construct sequential classifiers. Markov models and Nth-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 based on these series. N-gram methods include two sub-methods: point-based and path-based. Point-based prediction makes the prediction based only on the last visited URL. In contrast, path-based predictions use more than one page as the observation in order to make a prediction. In many applications, point-based predictions cannot make very accurate prediction since it neglects the observed past visited page information to discriminate the different access patterns. As a result, path-based predictions are more popular. In this area, there is a question on how to choose the best 'N' for N-grams. Su et al. (2000) performed an empirical study 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. Su et al. (2000) also 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. Schechter et al. (1998) proposed a small variant version of the longest match method by defining a threshold to go down a certain sequential path. Pitkow and Pirolli (1999) suggested a way to make predictions based on *K*th-order Markov models. Because they prefer longer paths more than shorter ones, their algorithm has the shortcoming that the longer path are more rare in the web log history, thus the noise in longer paths could be higher than in shorter paths. This can result in the undesirable effect of reduced accuracy. Pitkow and Pirolli (1999) also proposed a way to identify patterns of frequent accesses, which they call the longest repeating subsequences. They then used these sequences for prediction.

3. Rule representation methods

3.1. Web logs and user sessions

Consider the Web log data from a NASA Web server shown in figure 1. 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.

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 in the log that are retrieved automatically after accessing requests to a document containing links to these files. 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):

```
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
```

Figure 1. Example web log.

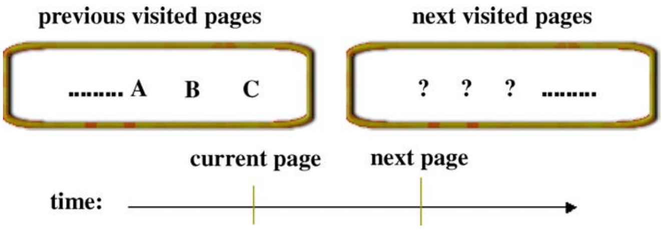


Figure 2. Moving window illustration.

W1			W2	
A1	A2	A3	P1	P2
A	B	C	A	C
B	C	A	C	D
C	A	C	D	G

Figure 3. A portion of the log table extracted by a moving window pair of size [2, 2].

(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 constant large gap in time interval as indicators of a new session. For example, for NASA data, the gap is 2 hours.

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 *W1*, and the consequent window as *W2*. 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. Figure 2 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. This table will be referred to as the *Log Table*, which represents all sessions in the web log. Figure 3 shows an example of such a table corresponding to the sequence (A, B, C, A, C, D, G), where the size of W1 is three and the size of W2 is two. In this table, under W1, A1, A2 and A3 denote the locations of the last three objects requested in the antecedent window, and P1 and P2 are the two objects in the consequent window.

3.2. Prediction rule representation methods

We now discuss how to extract sequential association rules of the form LHS → RHS from the session table. Our different methods below will extract rules based on different

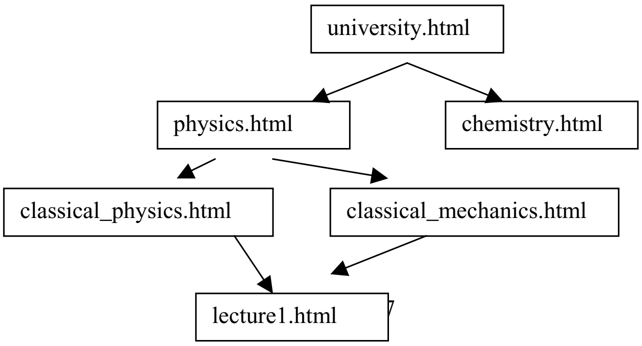


Figure 4. A web site example to illustrate rule-representation methods.

criteria for selecting the LHS. In this work, 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 → Uk where the pair {LHS, Uk} occurs most frequently in the rows of the table among all Ui's in the set {U1, U2, . . . , Un}. Ties are broken arbitrarily. This is the rule with the highest support among all LHS → Ui rules.

To illustrate, consider a real-world example in figure 4 of a student browsing a university's web site to look for some lecture material. Suppose the web site has several levels in a hierarchy of pages: where *university.html* is a parent page for *physics.html*, which in turn is a parent of *classical_physics.html.* Finally, under *classical_physics.html*, the lecture notes page lectures.html is a parent of lecture1.pdf—the target for the student. The student in this example leaves a user session in the web log:

university.html, physics.html, classical-physics.html, lectures.html, lecture1.pdf

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

$$\text{sup} = \frac{\textit{count}(\text{LHS, RHS})}{\textit{count}(\textit{Table})} \tag{1}$$

$$\textit{conf} = \frac{\textit{count}(\text{LHS, RHS})}{\textit{count}(\text{LHS})} \tag{2}$$

In the equations above, the function *count*(*Table*) returns the number of records in the log table, and *count*(LHS) returns the number of records that match the left-hand-side LHS of a rule.

We now describe different rule-representation methods. The first method we consider is called the *subset rules*. These rules motivated by the traditional association rules that simply ignore the order and adjacency constraint between accesses (Agrawal and Srikant, 1994). Thus, when the association rules for transaction data method (Agrawal et al., 1996), are applied to the log table, we obtain these rules. To illustrate, consider an example shown in Table 1, where we require that the minimum support of the rules be 100%. The subset rules satisfying this minimum support requirement are shown on the right hand side of the table.

Table 1. Subset rules with minimum support of 100%.

W1	W2	Subset rules
A, B, C	D	$\{A, B, C\} \rightarrow D$, $\{A, B\} \rightarrow D$, $\{B, C\} \rightarrow D$, $\{A, C\} \rightarrow D$,
B, A, C	D	$\{A\} \rightarrow D$, $\{B\} \rightarrow D$, $\{C\} \rightarrow D$

Returning to the student-browsing example, a subset rule might be:

$$\{\textit{university.html, classical\_physics.html}\} \rightarrow \textit{lecture1.pdf}$$

Subset rules reflect the intuition that sometimes the relative order of pages is not important in future-page prediction. For example, suppose that a page *classical_mechanics.html* is another child of *physics.html*, which in turn is another parent of the page *lecture1.pdf.* In this case, the subset rule $\{\textit{classical\_mechanics.html, classical\_physics.html}\} \rightarrow \textit{lecture1.pdf}$ does not have any imposed order on the LHS. It will match any pattern in which a user browse *classical_mechanics.html* and *classical_physics.html* in any order before browsing *lecture1.pdf*

The second rule representation is called subsequence rules, which takes into account the order information in the sessions. A subsequence within the antecedent window is formed by a series of URLs that appear in the same sequential order as they were accessed in the web log data set. However, they do not have to occur right next to each other, nor are they required to end with the antecedent window. When this type of rules is extracted from the log tables, the left hand side of the rules will include the order information. Table 2 shows an example of subsequence rules under the minimum support requirement of 100%.

A subsequence rule has the same form as a subset rule with the constraint that the order of the items in the LHS of the rule has to be kept when matching a rule against a sequence of observations. Thus, the subsequence rule $\{\textit{university.html, classifical\_physics.html}\} \rightarrow \textit{lecture1.pdf}$ cannot match a user session in which *classical_physics.html* is viewed before *university.html*. Intuitively, the subsequence rules build more constraints in its representation and thus may be more accurate in prediction. This representation is motivated by the body of work in sequential association rule mining [P + 01].

The third rule representation is called the latest-subsequence rules. These rules not only take into account the order information, but also the recency information about the last page in the subsequence. In this representation, only the subsequence ending in the current time (which corresponds to the end of the window W1) qualifies to be the LHS of a rule. For example, Table 3 shows the latest-subsequence rules. In our student-browsing example, a

Table 2. Subsequence rules with minimum support of 100%.

W1	W2	Subsequence rules
A, B, C	D	$\{B, C\} \rightarrow D$, $\{A, C\} \rightarrow D$, $\{A\} \rightarrow D$, $\{B\} \rightarrow D$, $\{C\} \rightarrow D$
B, A, C	D	

Table 3. Latest subsequence rules with minimum support of 100%.

W1	W2	Latest subsequence rules
A, B, C	D	{B, C} → D, {A, C} → D, {C} → D
B, A, C	D	

latest subsequence rule might be

classical_physics.html, lectures.html → lecture1.pdf

The fourth rule representation is called the *substring rules*, which takes into account the order and adjacency information embedded in the sessions. Substrings are any sequence of adjacent URL's in *W1* starting anywhere in *W1*. They are taken as the LHS of rules. When these rules are extracted from the log tables, the left hand side of the rules must encode string information. For example, Table 4 shows the substring rules.

The substring rules are stricter than the subsequence rules. In the student-browsing example, a substring rule might be:

physics.html, classical_physics.html, → lecture1.pdf

The fifth representation is called the *latest-substring rules*. The "latest-substrings" are in fact the suffixes of the strings in W1 window. 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 substring ending in the current time (which corresponds to the end of the window W1) qualifies to be the LHS of a rule. These are also known as hybrid n-gram rules in some literature (Pitkow and Pirolli, 1999; Su et al., 2000). For example, Table 5 shows the latest-substring rules example.

In the student-browsing example, a latest substring rule is:

classical_physics.html, lectures.html → lecture1.pdf

Table 4. Substring rules with minimum support of 100%.

W1	W2	Substring rules
A, B, C	D	{A} → D, {B} → D, {C} → D
B, A, C	D	

Table 5. Latest-substring rule with minimum support of 100%.

W1	W2	Latest substring rule
A, B, C	D	{C} → D
B, A, C	D	

Viewed from another angle, latest-substring rules could also be considered as the union of Nth-order Markov models, where N covers different orders up to the length of *W1*. Therefore, it is more general than the N-gram models or Nth-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 experience a monotonically increasing precision curve.

When building a prediction model using one of the above rules, we wish to know which ones give the best performance. The subset rules correspond to the direct application of association rules, taking the user sessions as transaction data. This type of rules impose little constraint on the LHS and are very flexible in rule matching, and have been extensively studied in the literature (Agrawal and Srikant, 1994). Subsequence rules correspond to association rules for frequent patterns in sequential data (Agrawal and Srikant, 1995). The latest-substring rules corresponds to suffix rules that impose maximum order and recency constraints. Each method has their pros and cons, which we wish to study in this paper.

For all rule representation methods we defined above, we 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. This default rule corresponds to a zeroth-order Markov model. The main reason for adding the default rule in our study is to have a uniform recall of 100% for all models. This allows us to compare the different models on an equal footing based on accuracy, although doing so may reduce the accuracy of overall prediction (to less than 50%, for example). However, in an actual application a user may choose to remove the default rule and increase the accuracy of prediction. In the web log data, for example, a default rule can simply be the most frequently requested page in the training web log data.

4. Rule-selection methods

In classification, 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 this section, we will study different methods for rule selection. 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 W2. For N different test cases, let C be the number of correct predictions. Then the precision of the classifier is defined as

$$precision = \frac{C}{N} \tag{3}$$

4.1. Longest match

The longest-match method chooses a rule with the longest LHS that matches an observed sequence from all rules whose support value is above a minimum support threshold value. For example, suppose that for a testing case and for an antecedent window of size four, the observed sequence of URL's is (A, B, C, D). Suppose that from a rule set containing all rules whose support is above a minimum threshold value, we can find three rules that can be applied to this sequence:

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

In this case, the lengths of rule 1, rule 2 and rule 3 are four, two and one, respectively. Since Rule 1 has the longest length, the longest-match method will choose Rule 1 as the best rule and page E will be predicted to be accessed after D.

The rationale of the longest-match method is that longer surfing paths that also have high enough support values contain more accurate and richer signature information about the user-access patterns. This rule selection method has been extensively used in computer network area (Pitkow and Pirolli, 1999; Schechter et al., 1998).

4.2. Most-confident selection

In association rule mining area, a major method to construct a classifier from a collection of association rules is the most-confident selection method (Liu et al., 1998). The most-confident selection method always chooses a rule with the highest confidence among all the applicable association rules, among all rules whose support values are above the minimum support threshold. 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 4, an observed sequence is (A, B, C, D). Using the most-confident rule selection method, we can find 3 rules which can be applied to this example, including:

Rule 1: (A, B, C, D) → E with confidence 30%
Rule 2: (C, D) → F with confidence 60%
Rule 3: (D) → G 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 might also show a higher precision in the testing data.

4.3. Pessimistic selection

Both the most-confident and longest-match selection methods attempt to select a good rule among those rules that satisfy a minimum support criterion. However, setting the minimum support criterion is a difficult task. If the minimum support is set too high, it might miss some useful rules. If it is set too low, the prediction algorithm might suffer from overfitting, where the learned model only performs well on training data set but not on the entire data set. In this section, we introduce a method to combine both support and confidence of a rule into a single criterion known as the pessimistic criterion to avoid the need to set a minimum support value by human.

We can build a new selection criterion by combining the confidence and support for a rule to form a unified selection measure, based on the observed error and on the support for each rule, thus avoiding having to specify a minimum support value artificially. Consider a rule R1 = LHS → RHS, for which the count (LHS) $= K$ and the observed error rate is f; for this rule there are K training instances that support it. The observed error rate is simply the number of incorrect predictions divided by K. Also consider a random variable for modeling the true error rate. A random variable X with zero mean lies within a range of $2z$ with a confidence of $\Pr[-z \leq X \leq z] = c$. From normal distribution, the value of z can be obtained for any value of C. For example, $\Pr[-1.65 \leq X \leq 1.65] = 90\%$. For the above notation, we can set the random variable X to be $X = \frac{f-e}{\sqrt{e(1-e)/K}}$ where f is the observed error rate and e is the mean.

Based on the above formula, we can find the range of true error rate e for the rule R1 based on the observed error rate f and the number of supporting instances K. Let the given confidence range value be z, the confidence value corresponding to z be cf, the number of supporting instances count (LHS) be K, and the observed error rate be f. Then the upper bound on the estimated error e is Quinlan (1993):

$$U_{\mathrm{cf}}(f, K) = \frac{\left(f + \frac{z}{2K} + z\sqrt{\frac{f}{K} - \frac{f^2}{K} + \frac{z^2}{4K^2}}\right)}{\left(1 + \frac{z^2}{K}\right)}$$

This pessimistic-error estimate is then used as a rule-selection criterion, much in the same way the same criterion is used for pruning decision tree nodes (Quinlan, 1993). In particular, for a given confidence level, we can find the corresponding confidence limit z from the normal distribution. For example, for confidence $c = 95\%$, we have $z = 1.28$. Then we can select rules with whose $U_{\mathrm{cf}}(f, K)$ value is the smallest. Conversely, we can define a pessimistic confidence value for the rule as $1 - U_{\mathrm{cf}}(f, K)$, and choose the rule with the largest such value.

Intuitively, in pessimistic selection, we only use the upper limit of the error rate as the estimate on potential error rate in test data, because this method is always *pessimistic* about the accuracy of classification model. Therefore, it always expects a higher error rate using the classifier on unknown testing data. For a rule with a small support measure, K will be small, and the corresponding pessimistic error $U_{\mathrm{cf}}(f, K)$ will be large. Overfitting is naturally taken care of without imposing an artificial minimum support threshold.

As an example, consider the following rules:

Rule 1: (A) → B with confidence 100%, $K = 1$, $E = 0$,
Rule 2: (D) → G with confidence 80%, $K = 100$, $E = 20$

Here K is the support count for the rule, and E is the number of incorrect classifications on training data. Suppose that Rule 2 has been applied to predict on 100 cases in the training data set, of which 20 are incorrectly predicted. For a confidence level 75%, the estimated upper limit (or pessimistic limit) of the real error rate is $U_{0.25}$ (0.2, 100). Computing the pessimistic confidence on both rules, we get:

For Rule 1: pessimistic confidence $= 1 - U_{0.75}$ (0, 1) $= 25\%$,
For Rule 2: pessimistic confidence $= 1 - U_{0.75}$ (0.2, 100) $= 76.57\%$

The pessimistic-selection method picks a rule with the *highest* pessimistic confidence in all the applicable rules. Ties are broken arbitrarily. In this case, Rule 2 is regarded as more reliable. Thus, Rule 2 is considered to avoid overfitting much better than Rule 1, and is selected with G as the prediction.

5. Model comparison

In the previous sections, we presented five rule representation methods (subset, subsequence, latest-subsequence, substring and latest-substring) and three rule selection methods (longest match, most-confident selection and pessimistic selection). Each rule-representation/rule-selection pair gives rise to a prediction model. A question arises as to which model is the best in making correct predictions. In this section, we empirically compare these methods and analyze their pros and cons.

5.1. Experimental setup

Our goal is to select the best rule representation and rule-selection combination among all rules-representation and rule-selection methods. For rule representation, we have so far discussed the subset, subsequence, latest-subsequence, substring and latest substring rules. For rule selection methods we have discussed the longest match, the most-confident-selection and pessimistic-selection methods.

In order to perform the comparison, we employ three real data sets. CSSFU (School of Computing Science in Simon Fraser University) data contains 3 days' HTTP requests to the Apache Web server serving www.cs.sfu.ca domain. The log was collected from May 1st, 2001 to May 3rd, 2001. In this period there were totally 45,637 requests. There are a total of 4,682 unique visiting IP addresses, and 5,650 sessions, and 14,664 unique pages are requested. EPA (United States Environmental Protection Agency) data contains a day's worth of all HTTP requests to the EPA WWW server located at Research Triangle Park, NC. The log was collected from 23:53:25 on Tuesday, August 29 1995 to 23:53:07 on Wednesday, August 30 1995, a total of 24 hours. In this period there were totally 47,748 requests. There

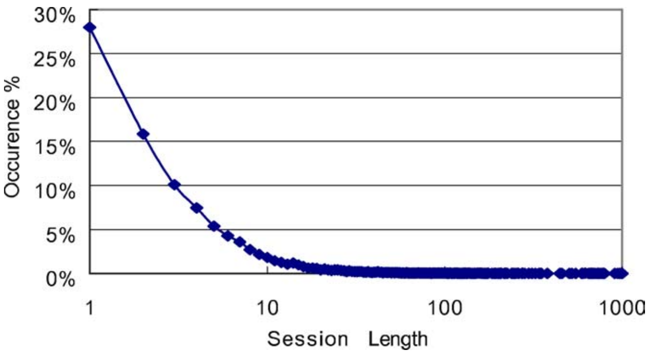


Figure 5. Session length distribution in NASA data.

are a total of 2,249 unique visiting IP addresses, 2,520 sessions, and 3,730 unique pages are requested. The NASA data is from NASA Kennedy Space Center WWW server in Florida. 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 totally 1,569,898 requests. There are a total of 72,151 unique IP addresses, having a total of 119,838 sessions. A total of 2,926 unique pages are requested.

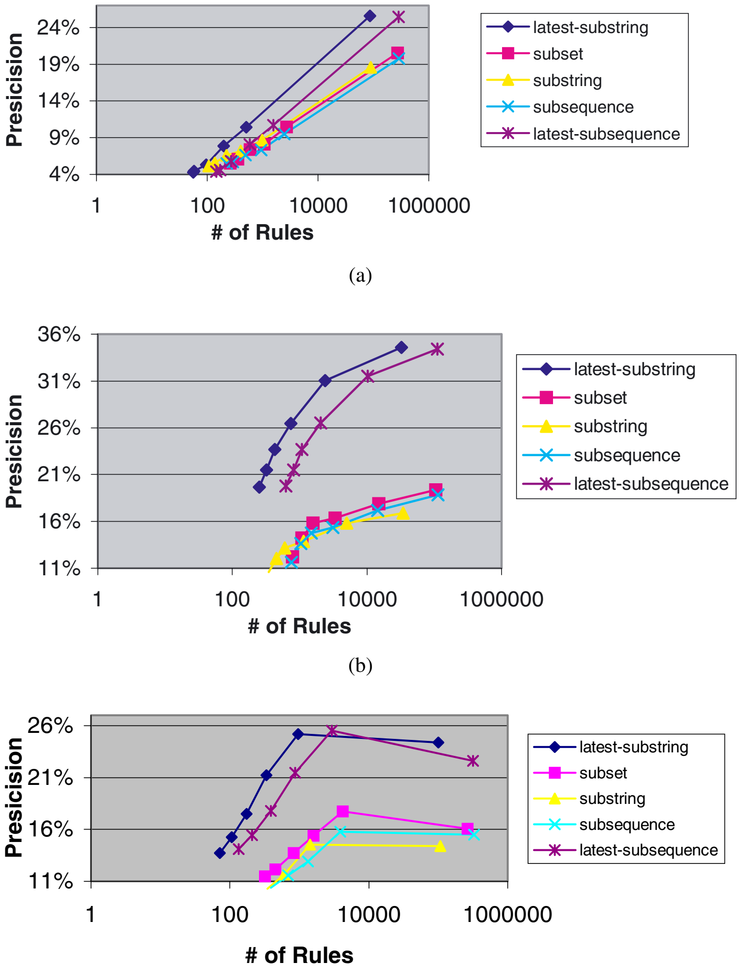
Before doing the experiments, we removed all dynamically generated content such as CGI scripts. We also filtered out requests with unsuccessful HTTP response code. For each data set, we split the web log into two parts, a training part for rule construction, and a testing part for evaluation. Figure 5 shows the session length distribution.

5.2. *Comparing rule-representation methods*

Figures 6(a)–(c) show precision comparison of different rule-representation methods under the longest match rule selection method, when the minimum support value varies between 0.0% and 0.1% with an interval of 0.02%. The X-axes show the number of rules generated corresponding to each minimum support value. Using the number of rules instead of the minimum support values directly to illustrate the variation of the prediction model is, in our opinion, a more direct way to demonstrate the system's performance.

The experiments were conducted on all three datasets (NASA, EPA and CSSFU). This and subsequent experiments use different sizes of training data and testing data of the three web logs. To ensure fair comparison, we use a default rule for all models under comparison. The default rule has an empty LHS, and a most popular page on the RHS. The use of this rule ensures that all models make predictions on all observations in the test data, thus the comparison is on which model can make the most correct predictions on the test data. However, the use of the default rules also "drags down" the overall precision of all models.

We would like to evaluate our models using a similar method to cross validation in machine learning. To do this we have to take into account the special property of the web logs, in that time only flows forward. Thus, we can take a segment of the web log as training



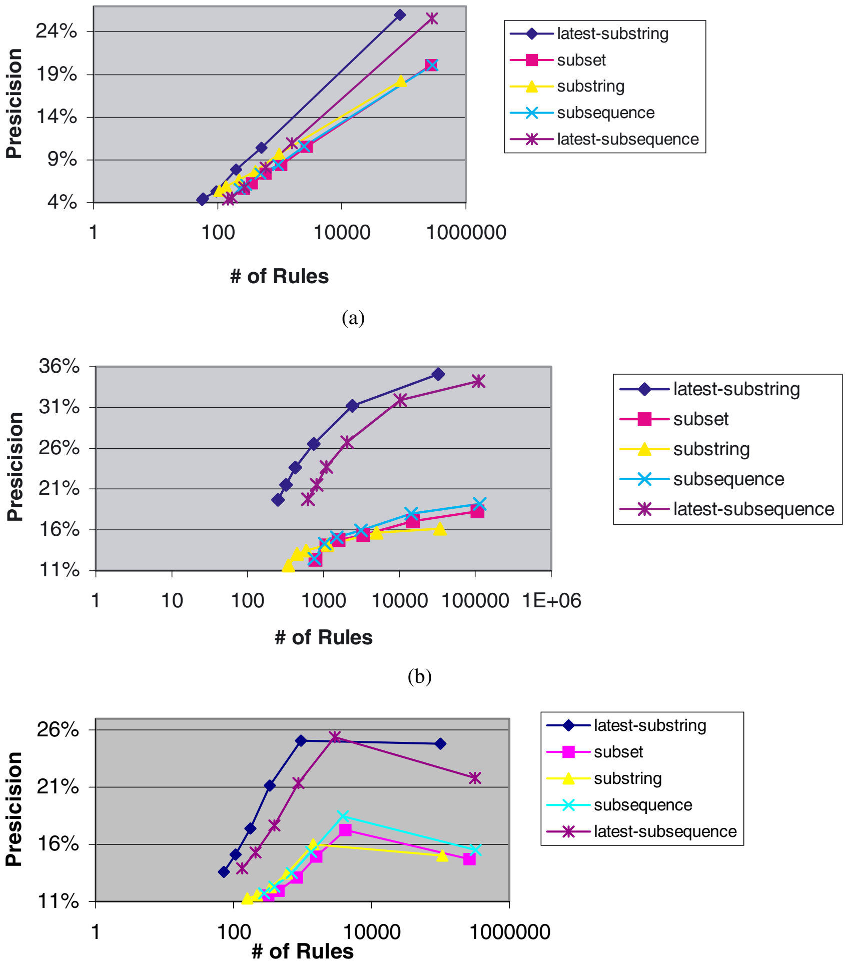
(a)

(b)

(c)

Figure 6. (a) Longest-match selection with the CSSFU data. (b) Longest-match selection with the EPA data. (c) Longest-match selection with the NASA data.

data and the next segment as testing data. We have observed similar experimental results for different sized training and testing data. Thus, due to limited space available for discussion, we only present the results derived from training data obtained from a segment of 100,000 URLs from each web log and the testing data obtained from the next segment of 25,000 URLs from the web logs. In the NASA data, the training data corresponds to 26 consecutive days of web log and the testing data corresponds to the subsequent 5 days of web log data. In each experiment, we vary the minimum-support threshold value for the experiment, resulting in rule sets with different number of rules. The precision of all methods is plotted



(a)

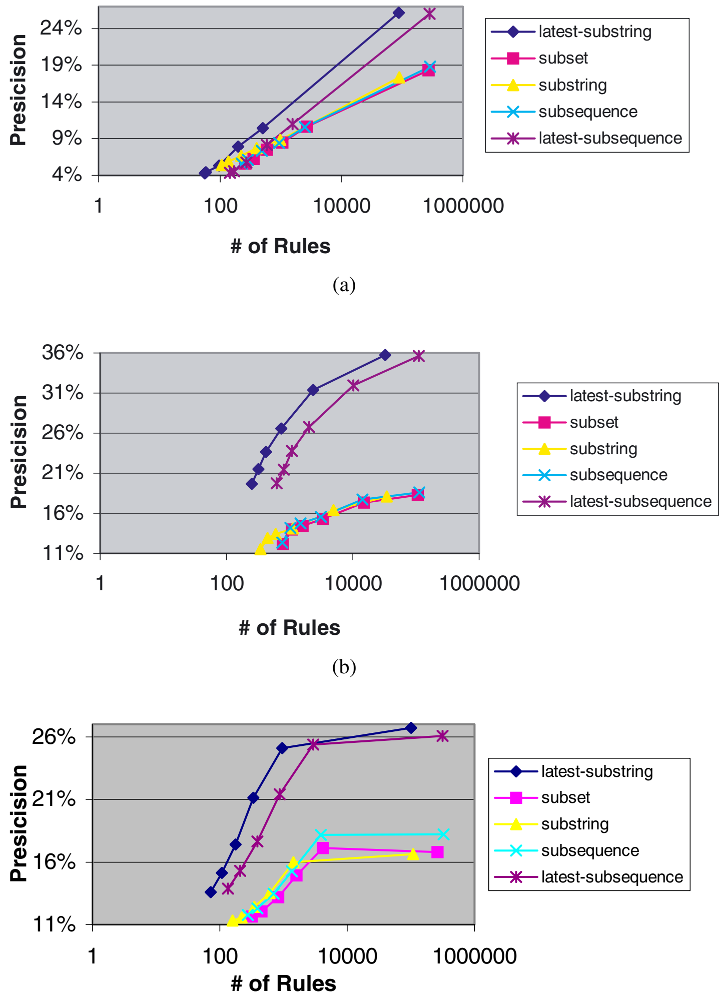
(b)

(c)

Figure 7. (a) Comparing rules with most-confident selection method using CSSFU data. (b) Comparing rules with most-confident selection method using EPA data. (c) Comparing rules with most-confident selection method using NASA data.

against the number of rules in each model, ranging from 0 to 100,000. Different models are then compared on the same graph.

Figure 7(a)–(c) show our comparison of the five rule representations under the most-confident selection method. As can be seen from the figures, the latest-substring representation dramatically outperforms the subset and the substring representations. We also note that after the number of rules reaches a level near 1,000, the precision of all rule-representation methods decline. This is a classical example of overfitting, where more specific rules actually degrade the performance.



(a)

(b)

(c)

Figure 8. (a) Comparing rules with pessimistic selection using CSSFU data. (b) Comparing rules with pessimistic selection using EPA data. (c) Comparing rules with pessimistic selection using NASA data.

Figure 8(a)–(c) show our comparison of the five rule-based methods under pessimistic selection method. The precision of all methods is plotted against the number of rules in the classifier, ranging from 0 to 100,000. As can be seen, the latest substring method dramatically outperforms the subset method and the substring methods. Because the pessimistic method can take care of the overfitting problem much better than the most-confident selection method, when the number of rules increases the latest-substring method still increased its precision.

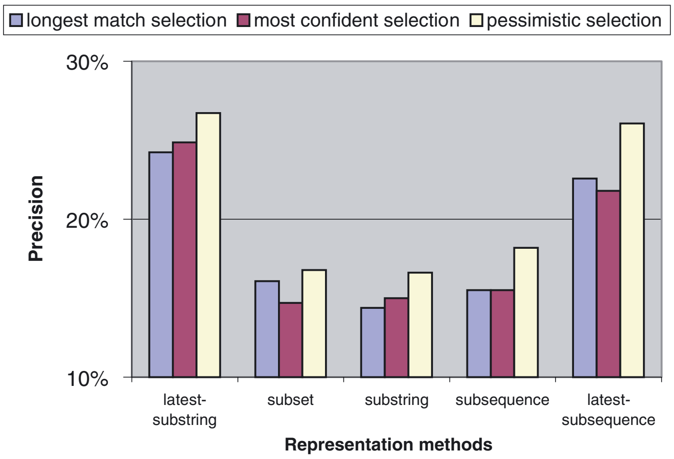


Figure 9. Comparison among rule-selection methods (NASA data).

5.3. *Comparing rule-selection methods*

From the above experiments, we conclude that the latest-substring method produces the best result among all the representation methods. In this section, we are going to compare different rule-selection methods, and suggest the best one.

In the next experiment, we compare the precision of different rule-selection methods under each rule-representation methods. In this experiment, we used 100,000 web requests in NASA web log as training data, and the next 25,000 web requests as testing data. As stated before, In the NASA data, this training data correspond to 26 consecutive days of web log data and the testing data correspond to the next 5 days of web log data. Figure 9 shows the comparison results. As can be seen from the figure, the pessimistic selection method is always the winner among the three rule-selection methods.

Given our earlier result that the latest-substring method performs best among all five rule-representation methods, we would like to explore this representation in more depth. Our next experiment compares all three rule-selection methods under the same latest-substring representation. The result is shown in figure 10. For this experiment, the training data consists of NASA web log with different sizes, ranging from 100,000 to 300,000 web accesses. The testing data is also NASA web log with the size of 1/4 of the training data. As can been seen, pessimistic-selection method always performs the best, and the performance gain increases with the size of training data.

5.4. *Analysis of the rule-representation methods*

From the above experiments, we can see that the combination of the latest-substring representation gives the best prediction precision among all methods. Here we will explain why this is the case.

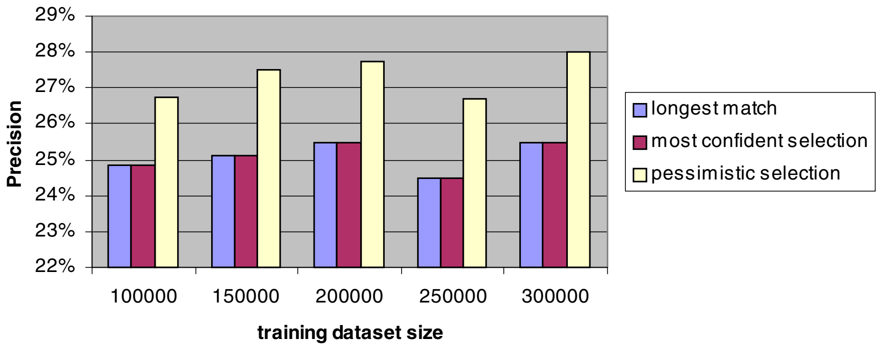


Figure 10. Comparison with latest-substring rules (NASA data).

W1				W2
A	B	C	D	E
B	C	D	E	F
C	D	E	F	G

Figure 11. Example log table extracted from a sequence of visits.

Consider an example sequence in the training data. Let a user session be

A, B, C, D, E, F, G

Now consider a moving window algorithm with *W1* = 4 and *W2* = 1. We can then extract the following log table from this sequence as shown in figure 11.

Now, consider rules with the same LHS "D". If we extract subset rules, these rules and associated confidence values are listed below:

Rule 1: D→E, Confidence = 33%
Rule 2: D→F, Confidence = 33%
Rule 3: D→G, Confidence = 33%

Given any observed case containing an URL "D", all three rules are equally good and there is no differentiation between them. Therefore, an arbitrary application of these rules will likely to result in an error. The same argument also holds for substring method.

On the other hand, suppose that we extract latest-substring method. Again we focus on the LHS = "D". In this case, no matter how large the *W1* window is, we always get only one rule

D→E, Confidence = 100%

Therefore, for a given test case, *E* will be predicted if the *W1* window ends with *D*.

Rule Rep/Methods	Longest Match Selection	Most Confident Selection	Pessimistic Selection (Best rule-selection method)
Subset rules			Second
Subsequence rules			Second
Latest-subsequence rules			Second
Substring rules			Second
Latest-substring rules (Best rule-representation method)	Second	Second	Overall winner!

Figure 12. Conclusion of the analyses: the shaded area indicates the winners in the comparison matrix.

This example shows that there are fewer latest-substring rules than either the subset or substring rules. In general, for a given *W1* window, there are *W1* latest-substring rules that can be extracted from the window. However, there are $2^{W1} - 1$ possible subset rules and $W1 * (W1 + 1)/2$ substring rules that can be extracted from the window. This large number of rules greatly reduces the confidence of each individual rule, and contributes to noise during rule selection process.

We attribute this phenomenon to the fact that the latest-substring rules encode important domain knowledge that the later URL's are more indicative of the next ones to come. We can similarly explain the poorer performance of the longest-match selection method. Although it makes sense that longer paths are more trustworthy in their predictions, such long paths are very rare in testing data. For example, figure 11 shows a session length distribution in the NASA log file, demonstrating exponential decay in the number of long sessions. In the end, a larger proportion of shorter rules apply to the test data than longer rules. The behavior of the shorter rules is similar to most confident selection method under the subset or substring representation. Therefore, longest-match does not perform as well as expected when compared to pessimistic selection.

We conclude that the latest substring representation coupled with the pessimistic-selection method gives the best prediction performance, as shown in figure 12.

6. Conclusions and future work

In this paper we studied different association-rule based methods for web request prediction. Our analysis is based on a two dimensional picture. In one dimension, we have a spectrum of rule representation methods, ranging from subset association rules to latest-substring rules. In the second dimension, we have the rule-selection methods, ranging from longest-match to pessimistic selection. In this matrix, we systematically studied the relative performance of different prediction models using real web logs as training and testing data. Our conclusion is that the method that uses the most domain knowledge and pessimistically estimates the error is an overall winner. In the future, we plan to compare more association rule methods and more rule selection methods. In addition, we wish to consider other type of domain knowledge to include in our rule representation.

Acknowledgment

We would like to thank NSERC and IRIS-PRECARN for their generous support for this research. We also thank Henry Zhang and Jia-wei Han for valuable comments.

References

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I. 1996. Fast Discovery of Association Rules. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, pp. 307–328.

Agrawal, R. and Srikant, R. 1994. Fast algorithm for mining association rules. In Proceedings of the Twentieth International Conference on Very Large Databases, pp. 487–499.

Agrawal, R. and Srikant, R. 1995. Mining sequential patterns. In Proceedings of the 1995 Int. Conf. Data Engineering, Taipei, Taiwan, pp. 3–14.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. 1984. Classification and Regression Trees. Wadsworth, Belmont, CA.

Liu, B., Hsu, W., and Ma, Y. 1998. Integrating classification and association mining. In Proc. of the Fourth Int'l Conf. on Knowledge Discovery and Data Mining (KDD-98), New York, pp. 80–86.

Mannila, H., Toivonen, H., and Verkamo, I. 1998. Discovering frequent episodes in sequences. In Proceedings of the First Int'l Conference on Knowledge Discovery and Data Mining (KDD'95), Montreal, Canada, AAAI Press, pp. 210–215.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal U., and Hsu, M.-C. 2001. PrefixSpan: Mining sequential patterns efficiently by PrefixProjected pattern growth. In. Proc. 2001 Int. Conf. Data Engineering (ICDE'01), Heidelberg, Germany, pp. 215–224.

Pitkow, J. and Pirolli, P. 1999. Mining longest repeating subsequences to predict World Wide Web surfing. In Second USENIX Symposium on Internet Technologies and Systems, Boulder, CO.

Quinlan, J.R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

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

Schechter, S., Krishnan, M., and Smith, M.D. 1998. Using path profiles to predict HTTP requests. In Proc. 7th International World Wide Web Conference, Brisbane, Qld., Australia, pp. 457–467.

Su, Z., Yang, Q., Lu, Y., and Zhang, H. 2000. Whatnext: A prediction system for web requests using N-gram sequence models. In Proc. of the First Int'l Conf. on Web Information Systems and Engineering Conference, Hong Kong, pp. 200–207.

Su, Z., Yang, Q., and Zhang, H. 2000. A prediction system for multimedia pre-fetching in Internet. In Proc. 2000 Int'l ACM Conf. on Multimedia, Los Angeles, California.

Wang, K., Zhou, S.Q., and He, Yu. 2000. Growing decision trees on association rules. In Proceedings of the 2000 International Conference of Knowledge Discovery in Databases, SIGKDD, pp. 265–269.

Qiang Yang is an associate professor at Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China. His research interest is in data mining for Web and CRM applications, automatic planning, and knowledge management using case based reasoning. He obtained his PhD from University of

Maryland in 1989. Prior to his current appointment, he was a faculty member at University of Waterloo and Simon Fraser University in Canada. He is an IEEE Member.

Ian T.Y. Li is an executive of a data mining company in China. He obtained his M.Sc. degree from Simon Fraser University in 2001. His interest is in data mining and its application to customer relationship management.

Ke Wang is an associate professor at Simon Fraser University in Canada. He obtained his PhD from Georgia Institute of Technology in 1986. Prior to his current position, he was a faculty member at the University of Singapore. His interest is in databases and data mining, including data mining applications on the Web, for customer relationship management and for biological applications.